Network Lifetime in Wireless Networks

Master Thesis Applied Mathematics

J. Verbree

October 11, 2007

Network Lifetime in Wireless Networks

Jan-Maarten Verbree

Network Lifetime in Wireless Networks

Jan-Maarten Verbree

Master Thesis
Enschede, October 11, 2007

University of Twente, the Netherlands
Faculty of Electrical Engineering, Mathematics and Computer Science
Department of Applied Mathematics
Chair of Discrete Mathematics and Mathematical Programming

Supervisors:
Dr. J.L. Hurink (University of Twente)
Dr. Ir. M. de Graaf (University of Twente, Thales)

Committee:
Dr. J.L. Hurink
Dr. Ir. M. de Graaf
Ir. A. te Marvelde (Thales)

Preface

During my internship at the Distributed Computing Group from the ETH in Zurich in 2006, I became more and more fascinated about wireless networks and their unsolved puzzles. For my final project there was again an opportunity to work on wireless networks, which I could not resist. The research question how to maximize the network lifetime comes from Maurits de Graaf, working at Thales Land & Joint Systems and at the University of Twente. During my research we bounded it more and more and finally focused on one specific algorithm, MPR flooding. This algorithm is used in the protocol OLSR, which is well known at Thales. Although the research is strongly related to Thales, I worked at the Discrete Mathematics and Mathematical Programming group at the University of Twente.

I would like to take this opportunity to thank some of the people who helped me during my final project. First of all, my supervisors Johann Hurink and Maurits de Graaf, whom I am very thankful for the many comments and many hours of discussion. The meetings we had could easily last two hours, even though we had meetings every two weeks. Their detailed comments, both with their own style, were useful for the improvement of my thesis. I also would like to thank the DMMP group for the pleasant time and making me feel one of the group already from the first day. In particular, I would like to thank my roommates, Jacob Jan and Maurice, for their interest in my research, their help and above all for the great atmosphere in the room!

Abstract

Mobile wireless networks offer numerous possibilities for applications in the current society. Since the radio transmitters become smaller, more and more mobile networks in which persons carry the mobile devices are developed. An example for such an network is given by a data communication network for soldiers or emergency workers, in which positions are exchanged in order to allow all persons to know the positions of their mates. Since each person has to carry its own communication equipment, every decrease in weight of the mobile communication equipment is useful. Since the energy supply, the batteries, forms a main part of the weight of the communication equipment, one aspect of the research to decrease the weight is concerned with developing highly energy-efficient communication algorithms that maximize the 'network lifetime'.

In this thesis, we explore the Network Lifetime Problem (NLP). The goal is to maximize the network lifetime, which is defined as the time until the first node runs out of energy. The first part of the thesis gives a classification methodology for algorithms and problems concerning the NLP. This methodology gives insight in the background of the NLP, proposed solutions for the NLP, and also information to compare algorithms fairly. Furthermore, some of the existing algorithms for the NLP are discussed. The main conclusion of the first part is that every algorithm is basically designed for one specific problem.

The second part of this thesis is devoted to Multipoint Relay (MPR) flooding and the ability of MPR flooding to maximize the network lifetime. MPR flooding is a technique that optimizes flooding by reducing the number of retransmissions. In a simple flooding approach all nodes have to relay a packet, but in MPR flooding only the selected MPRS. We present a study on MPRs, a study on MPR flooding and a study on MPR selection algorithms, instead of one global view on MPR flooding. One of the results show that in networks in which the nodes are positioned at a grid, the resulting set of selected MPRs of a non-border node is independent of the MPR selection algorithm. We also prove that in several networks the network lifetime cannot be improved and is fixed, independent of the used MPR selection algorithms, because of the definition of MPRs. Based on this result, we study the relation between the network structure and the effect of MPR flooding with a specific MPR selection algorithm on the network lifetime. We show that the performance of MPR flooding depends on a special property of the network, the so called 'Maximum Forcedness Ratio'. The idea behind the 'Forcedness Ratio' is also used in a proposed MPR selection algorithm, the Maximum Willingness and Minimum Forced MPR selection algorithm. This algorithms is comparable with the Maximum Willingness MPR selection algorithm, as used in OLSR, in terms of network lifetime.

Samenvatting

Mobiele draadloze netwerken bieden talloze mogelijkheden voor toepassingen in de huidige maatschappij. Omdat de radiozenders kleiner worden, worden er steeds meer mobiele netwerken ontwikkeld waarin personen de mobiele zenders met zich meedragen. Een voorbeeld is een datacommunicatie netwerk voor soldaten of hulpverleners, waarin onderling posities worden uitgewisseld. Aangezien iedereen zijn eigen communicatie-apparatuur moet dragen, is elke gewichtsvermindering nuttig. De energievoorziening, de batterijen, bepaalt een groot deel van het gewicht. Daarom is een onderzoeksaspect met betrekking tot de gewichtsvermindering het ontwikkelen van zeer energie efficiënte communicatie algoritmes die de netwerk levensduur maximaliseren.

In deze scriptie wordt het 'Network Lifetime Problem' (NLP) verkend. Het doel is om de netwerk levensduur (Network Lifetime) te maximaliseren. Deze is gedefinieerd als de tijd totdat een van de zenders geen energie meer heeft. In dit deel beschrijven we een classificatie methodologie voor algoritmes en problemen met betrekking tot het NLP. Deze methodologie geeft inzicht in de achtergrond van het NLP en voorgestelde oplossingen voor het NLP en biedt informatie om algoritmes eerlijk te kunnen vergelijken. Ook worden enkele bestaande algoritmes voor het NLP besproken. De hoofdconclusie van het eerste deel is dat elk algoritme wordt ontworpen voor een specifiek probleem.

Het tweede deel van de scriptie is gewijd aan Multipoint Relay (MPR) flooding en het vermogen van MPR flooding om de netwerk levensduur te maximaliseren. Flooding is het overspoelen van een netwerk met berichten. MPR flooding is een techniek die flooding optimaliseert door het aantal zenders dat berichten moet doorsturen te verminderen. In een simpele benadering van flooding moeten alle punten in een netwerk een datapakket doorsturen, terwijl bij MPR flooding alleen geselecteerde MPRs dit doen. In deze scriptie bestuderen we MPR flooding niet globaal, maar kijken we afzonderlijk naar MPRs, MPR flooding en MPR selectie algoritmes. We bewijzen dat in 'raster netwerken' een punt in het midden een vaste verzameling MPRs selecteert, ongeacht het gebruikte MPR selectie algoritme. We bewijzen ook dat in verschillende netwerken de netwerk levensduur van tevoren vast staat vanwege de MPR definitie. We laten vervolgens het verband zien tussen de netwerkstructuur en de netwerk levensduur voor MPR flooding met verschillende MPR selectie algoritmes. Uit de resultaten blijkt dat de 'Maximum Forcedness Ratio' (Maximale Geforceerdheid Ratio) van het netwerk, van invloed is op het prestatieverschil tussen de MPR selectie algoritmes. De 'Forcedness Ratio' wordt ook gebruikt in het nieuw voorgestelde 'Maximum Willingness and Minimum Forced' MPR selectie algoritme. Dit algoritme blijkt qua netwerk levensduur vergelijkbaar met het 'Maximum Willingness' MPR selectie algoritme, zoals gebruikt in OLSR.

Contents

Chapter 1

Introduction

1.1 Research Motivation

The last few years wireless networks have become very popular both in research and practical applications in all kind of domains. One specific field in which wireless networks are used nowadays is the military field. Military wireless networks are mainly used for communication and location exchange between soldiers and military vehicles. As a consequence, the soldiers have to carry their communication equipment with them. But the soldiers have to carry also weapons, ammunition and other equipment. Therefore, every decrease in weight of the mobile communication equipment is useful.

Besides the military field, mobile wireless networks in which locations and data are exchanged are also suitable for the emergency services. Take for example a forest-fire that has to be extinguished by fireworkers. In such situations it is of great importance that the fireworkers are placed at the right spots at each moment to control the fire as soon as possible. Therefore, the fireworkers have to me mobile and exchange their positions such that they can get the right instructions from the fire commander. Not only for fireworkers, but all kind of emergency workers can use such networks. But like the soldiers, the emergency workers also have to carry the mobile communication equipment besides their personal equipment.

A main source for the weight of the communication equipment comes from the energy supply, the batteries. Since these batteries also have to be carried by the soldiers or emergency workers, one aspect of research is to develop algorithms that are highly energy-efficient and maximize the network lifetime, i.e. the time until the first node runs out of energy. Using such algorithms may result in equipment where less batteries is needed. Besides the military and emergency service domain, maximizing the network lifetime is also useful for civil networks, like many sensor networks or networks formed by laptops in a meeting.

In this thesis we focus on the problem of maximizing the network lifetime by discussing the context of the problem and analyzing an existing algorithm. This algorithm is called MPR flooding and used in a wireless network protocol called OLSR which receives a lot of attention in the wireless network area.

1.2 Problem Statement

In an ideal mobile network, the network must work without user intervention. Therefore, the network has to be self-configurable and energy-efficient to avoid replacing batteries too often. The problem to have a network as long as possible running without replacing a battery, is often formulated by the goal to maximize the network lifetime. This thesis addresses this problem, called the Network Lifetime Problem (NLP). The NLP makes only sense in a multihop network, i.e. a network in which multiple nodes may be utilized for relaying communication traffic from a source node to a destination node. In such networks the network lifetime can be influenced by the selection of nodes that have to retransmit the message. However, in single hop networks, the source is the only node that transmits and there are no nodes selected to relay the message. As a consequence, only the sources of messages influence the network lifetime and since one normally assumes that the initiation of communication cannot be influenced, there is no possibility to maximize the network lifetime in a single hop network.

The first objective of this study is to explore the context of the NLP. We do not intend to give a complete overview of the existing work on the NLP, but present a classification methodology to characterize the existing algorithms and problems. In this context, we also discuss several existing algorithms for the NLP. The most important goal of this part is to give some feeling for the problems and existing work. It also can be seen as a widespread introduction to second objective of the thesis.

The developed classification methodology shows that different situations have their own specific problem to maximize the network lifetime. Therefore, the second objective is to look at one specific situation or algorithm and to analyze the existing results in order to improve the algorithm. The specific algorithm discussed in this thesis is called Multipoint Relay flooding.

1.3 Basic Definitions and Notations

Throughout the thesis we use several definitions that are not specifically related to one chapter or section. For clarity we introduce them already at this point.

Definition A graph $G(V, E)$ with a node set V and a set of undirected edges E is called a connected network graph if every two nodes of V are connected by a path.

Definition Assuming that each node has a geographical location, the distance $d_g(u, v)$ between two nodes u and v in V, is defined as the Euclidean distance between u and v.

Definition The hop distance between two nodes u and v in V is defined by the minimal number of edges on a path between u and v.

Definition The set of nodes $N^k(u)$ denotes the strict k-hop neighborhood of node u, i.e. the set of nodes for which the hop distance to u equals k.

1.4 Structure of the Thesis

The thesis is split into two parts, describing each one objective. The first part of the thesis is a survey of the NLP. Chapter 2 describes the classification methodology of algorithms and objectives in wireless networks. Next, we focus more on the NLP and discuss several classes of the NLP. The last chapter of Part I, Chapter 4, discusses algorithms related to the NLP.

Part II is completely addressed to Multipoint Relay flooding, a technique to optimize flooding by reducing the number of retransmissions, which is used in a popular protocol called OLSR. We study several facets of Multipoint Relay flooding and focus specifically on ability of the technique to maximize the network lifetime. The structure of Part II is described in Chapter 5.

The first part is written in a descriptive way to get introduced in an easy way to the several facets of the NLPs. The second part is written in a mathematical way in order to lay focus on the theorems and corresponding proofs.

Part I

Survey of the Network Lifetime Problem

Chapter 2

Classification of Algorithms for Wireless Networks

There are several reasons for developing and using a classification methodology for algorithms in wireless networks. First of all, due to the enormous number of such algorithms, classification of algorithms is helpful to get a clear overview of the types of available algorithms. Furthermore, it helps to get more insight in a specific algorithm. The classification also shows the necessary adjustments for an existing algorithm to be adapted in other wireless situations and is therefore interesting for scientific research to improve algorithms. Last but not least, classification is crucial to compare different algorithms fairly as each algorithm has its own assumptions and objectives.

This chapter has the following structure. The first section describes several types of wireless networks and possible network properties. Since many algorithms are designed for specific networks, this characterization of the wireless network is very useful in a classification methodology. Next, the possible variables of an algorithm are discussed in Section 2.2. By varying the variables the algorithms have the opportunity to satisfy several objectives as much as possible. These potential objectives for which an algorithm is designed are listed and discussed in Section 2.4. The final section of this chapter handles specific mechanisms of an algorithm.

In our classification methodology we reuse the methodology schemes and explanations developed by Wang and Xiao [50] for the assumptions and objectives. They focused primarily on energy-efficient scheduling mechanisms in wireless sensor networks. We focus more on general ad hoc networks and not specifically sensor networks. Our contribution lies in that we develop the methodology further by presenting additional classifiers and that we specify the mobility and transmission properties further by discussing existing mobility and transmission models.

2.1 Wireless Networks

A wireless network is basically nothing more than a set of nodes that form a connected network via wireless communication. Each node has a transmitter and can reach all nodes in its transmission area, which we call *direct* communica-

tion. If other nodes are utilized for relaying communication traffic from a source node to a destination node, it is called *multihop* communication. Many algorithms in the wireless network area use the multihop approach or are designed for multihop networks.

2.1.1 Types of Wireless Networks

Although there are many wireless networks developed in the last few years, only two major types of wireless networks exist: Cellular Networks and Ad Hoc Networks. We describe these types of networks and their corresponding tasks in the following.

Cellular Networks have stationary base stations with a certain transmission range that divide the communication field into cells. Mostly, cells overlap each other partly. Each node belongs to a specific cell and therefore to the corresponding base station. If a node is located in an overlap area, it generally belongs to the base station with the strongest signal. A base station is only responsible for the nodes that are in its cell and therefore it regards only nodes that enter or leave its cell. The communication between nodes in the same cell goes via the base stations, using a direct connection from the source to the base station and from the base station to the destination. To establish communication between nodes in different cells, also the base stations are used. First the source transmits communication traffic directly to the base station. This base station communicates to the base station to which the destination node belongs. Finally the base station of the destination communicates directly to the destination node.

The most important task of a cellular network is long distance communication between mobile devices, like mobile telephones. One main problem in this area, for example for GSM networks, is to guarantee a communication without disturbance and interruption, even when nodes moves from one cell to another. This problem is called *Cellular Handoff*.

Ad Hoc Networks are wireless networks in which the determination of which nodes have to forward data is dynamically based on the network topology. This in contrast to networks with designated forwarding nodes, like base stations, routers, switches and hubs. *Mobile Ad Hoc Networks (MANETs)* are a special type of wireless ad hoc networks, in which nodes are mobile. The key element of a MANET is that it is an self-organizing structure that allows nodes to join or leave the network, resulting in a continuously changing network topology. To join the network the newcoming nodes have to connect to at least one of the existing nodes in the network, otherwise the network will not become aware of the new node. MANETs use a multihop approach for communication between two non-neighboring nodes in the network.

MANETs are fundamental structures that can be used for other applications to build upon. The communication layer is primarily used for data messaging, like in computer networks. Finding multihop routes from one node to another is one of the main problems of ad hoc networks. The nodes in a MANET are mostly assumed to have limited battery capacity and therefore limited transmission range.

A special class of Ad Hoc Networks are *Sensor Networks*, i.e. networks in which nodes are equipped with special sensors. One can think of termal sensors, visual sensors, pressure sensors, acoustic sensors and so on. Mostly the sensors are very tiny and have very limited battery capacity and memory.

The main tasks of Sensor Networks are Data Gathering and Monitoring. *Data Gathering* means that all nodes have to gather data and spread it to some (designated) nodes that combine the data. Because of this, sensor networks are sometimes called data driven networks. *Monitoring* is more or less a special case of Data Gathering, as its objective is to observe certain aspects, for which obviously data gathering is necessary. Wireless Sensor Networks have many applications, like fire detection in forests, battlefield surveillance, interactive museums and vehicle tracking and detection. The combination of sensor capability and the self-organizing structure makes Sensor Networks also extremely suitable for monitoring in inaccessible terrain or disaster relief operations.

2.1.2 Network Properties

As mentioned in the introduction of this chapter, we reuse the classification scheme of Wang and Xiao. The scheme as used in [50] is given in Figure 2.1. Although they focused on sensor network, the main part of the listed design assumptions is also useful for general wireless ad hoc networks, sometimes with a little bit of interpretation. In the following, we describe shortly the meaning of several properties. We occasionally also discuss how some properties can be interpreted to be useful for ad hoc networks too. After the description list, we zoom in on two network properties: transmission area and mobility. Commonly used models are discussed, which can be used for further characterization of a wireless network and therefore also for an algorithm that is designed for a specific network.

Common Assumptions As wireless nodes work on battery energy, they have limited energy supply. Wang and Xiao assume the network to be needed for a long time, otherwise it is not needed to create special mechanisms to save energy, which is denoted by *Long Network Lifetime*.

Network Structure In a *flat* structure each node has the same functionality and role. In a *hierarchical* structure some nodes are designated special tasks, like collecting information or running an algorithm to detect the neighborhood structure. Cellular networks and also other cluster-based networks are typical examples of hierarchical structures.

Sensor Deployment There are many ways to deploy wireless transmittor nodes, including sensor nodes, varying from randomly (dispersing nodes from an airplane) to exactly (installing nodes at selected locations). Each situation has its own model that can describe the positioning. Dispersing can be modelled by a random distribution and the exactly positioned nodes can be described e.g. by a grid. The variation of deployment manners results in a enormous variation of these models. To determine whether the sensor density is high or normal, we look at the number of working sensors. We distinguish working sensor from redundant sensors, as the first actually contribute to the quality of an algorithm, but the latter can

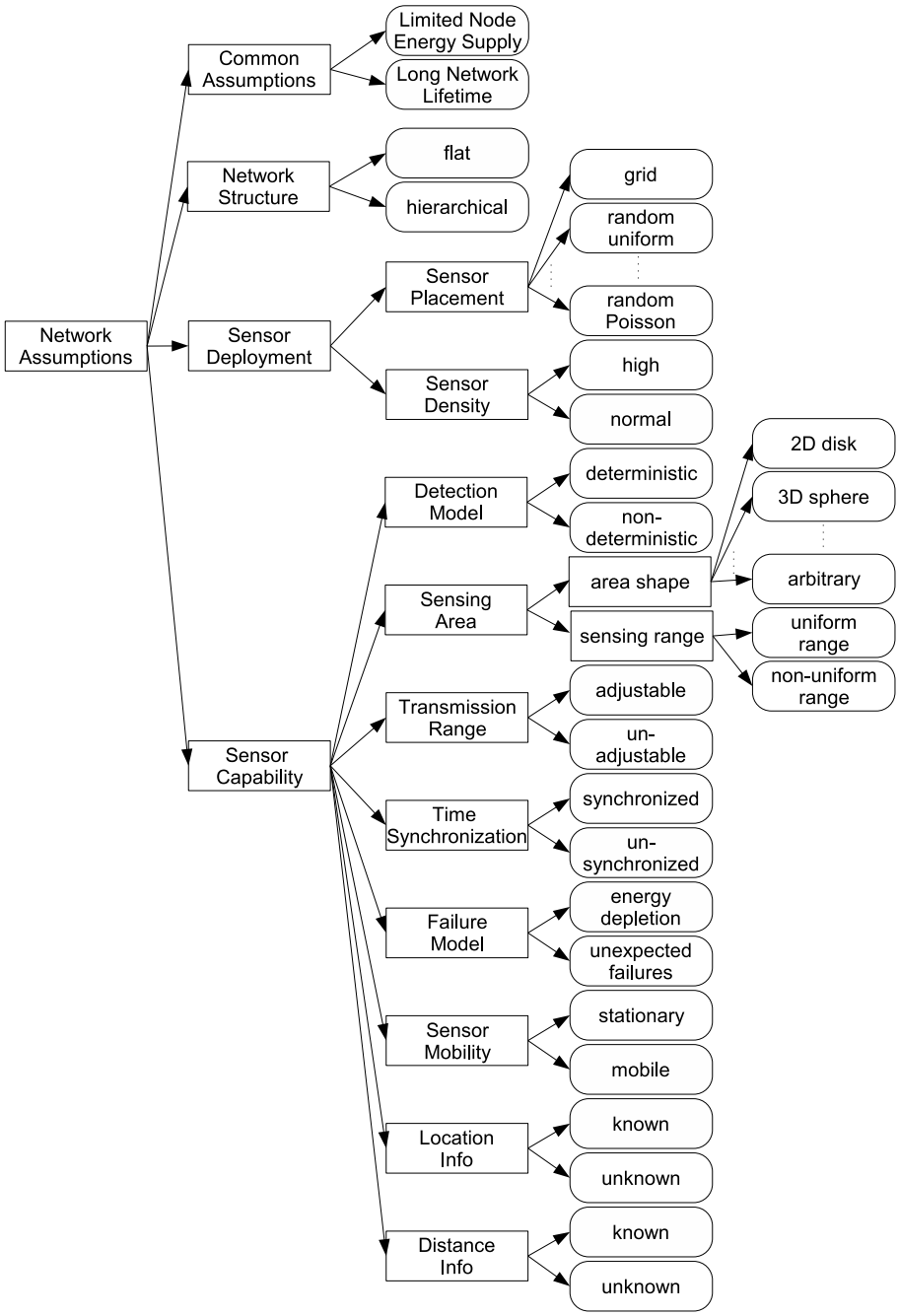


Figure 2.1: The property scheme for wireless networks. This is a slightly adjusted version of the scheme in [50].

be switched off without loosing quality of performance. If the total number of nodes is on the same order as the number of working sensors it is called a *normal* sensor density; if the total number of sensors is orders of magnitude higher than the working sensors it is called a *high* sensor density.

Detection Model In case of a deterministic model it is assumed that a node can detect every object in its sensing range. The non-deterministic variant leaves the possibility open that an object can not be detected, although it is inside the sensing range of a node. For non-sensor nodes this property can also be interpreted as a detection model for communication if a neighbor falls in its transmission area. We discuss this further later on.

Sensing Area In addition to the detection model, detecting objects is also dependent of the sensing area of the sensor nodes. For non-sensor nodes it is more useful to define the type of transmission area. This is studied in depth in after this description list, where several commonly used models for transmission areas in wireless ad hoc networks are discussed.

Transmission Range Many topology control algorithms assume an adjustable transmission power. This assumption is also one of the bases for the problem to maximize the lifetime of a network. The three distinct types of tranmission ranges assignment are: continuous, discrete and on-off. In the first case, every value between zero and the maximum transmission range can be chosen. Using discrete ranges, there are a fixed number of energy levels that result in certain transmitting ranges. A special case of the discrete transmission power assignment is the on-off situation in which only two possibilities are available: transmitting with full power or not transmitting at all.

Time Synchronization This feature is mostly used for waking up from sleep mode, but is also an aspect in some topology control algorithms. Time synchronization is then necessary in order to define time limited rounds and time stamps to topology control messages, see for example the algorithm in [57].

Failure Model In the real world a node can fail in many ways: it can run out of energy, have a temporarily interruption or even be destructed. Destruction is not unlikely to happen when the wireless network is used in the military field. The assumption with respect to node failure is important to manage the topology well, for example to avoid that a node that is interrupted temporarily is seen as a lost node.

Sensor Mobility In the classification scheme presented in Figure 2.1 only two options are specified, *stationary* and *mobile*. There is however a certain degree of mobility for each network, varying from low-mobile to high-mobile. For the mobility of nodes there exist also several models that describe mobility further, just like there exist models for tranmission. These models are presented at the end of this section.

Location Info The specific location information, i.e. the geographical location, can be used to determine the overlapping of sensing areas and to

compute the distance between neighbors. In stationary wireless networks where the nodes are placed at selected locations, the location information can be hard coded. Mobile situations and randomly distributed nodes need equipment like GPS or specific location algorithms to detect their position.

Distance Info If location information is available distance information can be computed easily; the reverse is however not true. Distance information can be used to compute the transmission power needed for sending a message to a neighbor node. A commonly used method to estimate distances is Received Signal Strength Indication (RSSI). RSSI is a measurement of the received radio signal energy, but does not say anything about the quality. Combining RSSI with a power attenuation model and some knowledge about the transmittor, an estimation can be done about the distance to the transmittor. Besides RSSI a method called Time Distance Of Arrivals (TDOA) exists. The distance estimation is based on the difference in time between or the time stamp and the arrival time, in case of a synchronized network, or the time of the message that is sent and the arrival of the reply message, in case of unsynchronized networks.

In the following we give some more detailed information on the Transmission Models and the Mobility Models. Since the topology of a network is determined by the connection between nodes it is important to look closer at the transmission model in a network. In general, identical nodes are assumed and therefore, often also identical transmission models for each node. We discuss the commonly used models shortly. For more information about used models in wireless sensor networks we refer the reader to [45].

Disk Graph A simple model for transmission in a network is a Disk Graph. The transmission area of each node is modelled as a disk with a certain size, which may differ per node. A node can reach every node that is insides its disk. The extension of this model to three dimensions is called ball graph.

Unit Disk Graph A special type of the Disk Graph transmission model is called the Unit Disk Graph (UDG) in which every transmission area is represented as a disk with unit size. As the transmitters are equal, the disks can be scaled to have radius one. An example of a UDG is shown in Figure 2.2 in which every two nodes that have distance less than one are connected. The unit version of the ball graph is called Unit Ball Graph (UBG).

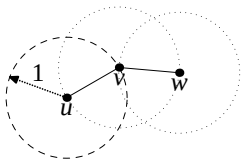


Figure 2.2: Example of a UDG. Node u is connected with node v, but not with node w.

General Graph The UDG model is a bit too idealistic: it assumes a flat world with no objects in the tranmission area of a node. It seems therefore not realistic to use UDGs to model the real world. To allow difference in heights and transmission disturbing objects, another model was created: the General Graph. This model only assumes the connectivy graph of multiple transmitters to be a general undirected graph. Although this model does not really conflict with reality, it neither gives a usable description of the transmission range and is therefore maybe too extreme.

Quasi Unit Disk Graph An intermediate model is the Quasi Unit Disk Graph. The tranmission range is described by two disks, one with radius R and one with radius $\rho \in (0, R]$, both centered at a node u. Nodes that are within the disk with radius ρ are connected with u, nodes outside the disk with radius R are not connected with u and nodes that are between those two disks may or may not be connected. Since the transmitters are assumed to have equal ranges, this model can also be scaled to a common unit: $R = 1$. This model is a combination of UDG and General Graph and makes it possible to use some geometric properties, but leaves enough possibilities open to make the model realistic.

Power Attenuation Power Attenuation is not a transmission model itself like UDG, but is a part of a transmission model, strongly related to the transmission range of a UDG. The power attenuation describes the decrease of power strength in an area and gives therefore additional information about the transmission model. In general it is assumed that the power needed to reach a node j from node i, denoted by $\mathbf{P}_{ij}$, dependents on the Euclidean distance between i and j, denoted by d_{ij}. The relation between the parameters is mostly approximated by the function: $\mathbf{P}_{ij} = d_{ij}^{\alpha}$. The value of α is called the path loss attenuation factor that usually is limited by $2 \leq \alpha \leq 4$.

Next, we discuss Mobility Models in more detail. Mobility means that the nodes are allowed to change position during time. Two categories can be distinguished: nodes that move in a determined way and nodes that move randomly. For the first category one can use the description of the determined movement of the nodes to model the mobility. We therefore look only at random mobility in wireless networks. First, we discuss the importance of mobility models. The rest of this section is used for the presentation of three mobility models.

Importance of Mobility Models Topology control in mobile situations is far more complex than topology control in stationary situations. This is because not only failure, but also the mobility has to be taken into account. The degree of mobility (from low-mobile to high-mobile) determines more or less the complexity and the number of failures. To give an idea of the performance of some algorithms that do not assume mobility on mobile networks we present Figure 2.3. This graphic from [57] shows the mean connectivity ratio, i.e. the ratio of connected node pairs to the total number of node pairs, of mobile networks for several topology control algorithms. The connectivity ratios are the results of 1000 simulations of 100 nodes with transmission range of $200m$ deployed randomly in an area of $900m$ by $900m$. The simulated algorithms are a local Minimum Spanning Tree (MST), the Relative Neighborhood Graph (RNG) and

Shortest Path Tree-based algorithms (SPT) with respect to minimum-energy. SPT-4 assumes a path loss attenuation factor $\alpha = 4$, and SPT-2 a path loss attenuation factor of $\alpha = 2$. More information on the algorithms and the simulation can be found in [57]. In the simulations all nodes send "hello" messages such that each node can detect its neighborhood. The interval between the transmitting of the "hello" messages is randomly selected by each node from $1 \pm 0.25s$. Every time that a node transmits a "hello" message, it also runs the topology control algorithm to create its local topology. Due to the combination of mobility and the "hello" message interval, it is possible that the algorithms use outdated or incorrect information which can lead to unconnected node pairs. If the nodes move faster, it is more likely to have incorrect information and therefore the results also show a decrease of the connectivity ratio for higher speeds. Note that the first values in the graphic are at a speed of $1m/s$ and not at $0m/s$. At a speed of $0m/s$ the connectivity ratio is one for each algorithm, but at low speed ($1m/s$) MST, RNG and SPT-4 already have low connectivity.

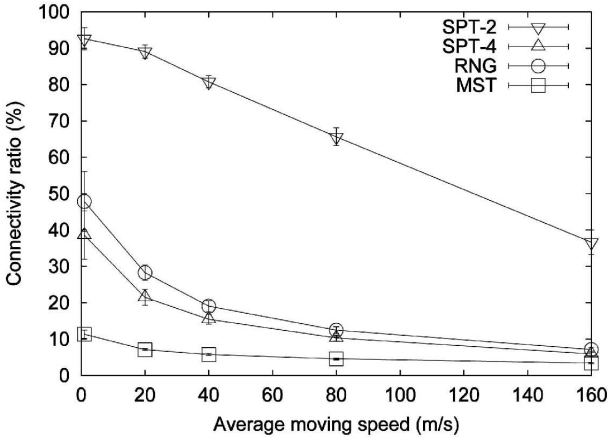


Figure 2.3: The connectivity ratios of the topology control algorithms SPT-2, SPT-4, RNG and MST in mobile wireless networks. The figure is taken over from [57].

As the consequences of mobility can be disastrous for an algorithm, the mobility assumption has to be taken into account seriously. Mobility can not just be ignored without consequences in the real world. The key aspects in which mobility influences the algorithm are [44]: increased message overhead and non-uniform spatial distribution of nodes. If nodes are stationary it is sufficient to detect only once where all the nodes are situated, but in mobile networks nodes need to inform neighbors about their new positions (if they know their positions) more often. This results in an increased message overhead. Therefore, in mobile situations algorithms are needed that require low overhead for setting up a routing structure. In many algorithms and simulations, a uniform spatial distribution of the nodes is assumed. It is doubtful if this assumption is realistic, especially in mobile situations.

Random Waypoint Mobility Model Mobility can be modelled in many ways, but the most used model in wireless ad hoc networks is the random

waypoint model ([20]). Each node chooses a destination or waypoint uniformly at random within some spatial boundaries in this model. The speed of the node towards this waypoint is chosen randomly at a uniform distribution in the interval $[v_{min}, v_{max}]$. After reaching its destination a node waits for a certain predefined time t_{pause} and moves then further to its new destination, prescribed by the same protocol. Although this model is simple and mostly used, is can have an unwanted effect: the *border effect*. As a node chooses a new destination, it is very likely that nodes that are at the border, will cross the center. This results in a high concentration of nodes in the center and low concentration of nodes at the borders. The intensity of the effect is determined by t_{pause}. The higher the value of t_{pause} the more uniform the distribution will be, as the border effect only occurs while nodes are moving. It is necessary to be alert for this effect, as when the nodes are close together less power is needed for transmission and can distort the real performance of an algorithm. The border effect however does not always have to be a drawback. In certain situations, like people moving in a city, the border effect can model some aspects in real life very well and one even wishes to achieve this effect.

Bettstetter Mobility Model The mobility model of Bettstetter ([2]) chooses randomly a direction and velocity from the uniform distribution in the intervals respectively $[0, 2\pi[$ and $[v_{min}, v_{max}]$. In case of a three dimensional model one could choose two angles independently in the interval $[0, 2\pi[$. The first value can be interpreted as the horizontal direction and the second value as the vertical direction. The time before choosing a new destination is usually taken randomly from an exponential distribution. In a similar way the velocity changes at certain times. If a bounded region is assumed, additional rules are needed to keep the nodes inside the bounded region.

Brownian Mobility Model A different approach is given by a Brownian model ([4]) in which there is no intentional motion. The moving process is totally described by probabilities like p_{stat} and p_{move}. The parameter p_{stat} describes the probability that a node remains stationary during the total simulation. In reality it is possible that a mobile node can not move anymore due to a broken motor for example, which can be modelled very well with this parameter. The parameter p_{move} represents the probability that a node moves at a given step. Going to zero, the network will become more stationary, going to one, the network will be more dynamic. If a node moves, it chooses its location for the next step randomly in a square with a certain side length. This model makes it also possible that a node moves out of a bounded region, for example the region in which they are deployed. If this is not desirable, one can add border rules to keep the nodes inside the deployment region ([2]).

2.2 Adjustable Network Variables

Algorithms are mostly designed to reach certain objectives. To reach this goal, they are allowed to influence or change some parameters or properties of the network. Some of the properties discussed in Section 2.1.2 are assumed to be fixed, i.e. these properties can not be changed by any algorithm, like the number of nodes or the battery capacity. But in general there are at least three aspects

that can be varied by an algorithm. These variables are the essence of the algorithms, as they influence the way of how a network operates. We present shortly the list of aspects that can be variables.

1. *Position:* Networks with a sensing task have to cover a certain area. If the position of the nodes can be controlled, the sensors can be instructed to move to specified positions in order to cover the complete area in an efficient way. Almost all algorithms with communication tasks like broadcasting do not assume that the location of the nodes can be controlled in such a way.

2. *Transmission Power:* By adjusting the power of a node, the transmission area can be increased or decreased. Increasing the area has the intention to communicate with more neighbors in a direct way, but uses more energy. Decreasing the transmission area saves energy, but has effect on the direct communication links. In general, more multihop transmission are needed if the transmission power is low instead of high. Power adjustment is a big topic of research in the wireless network community nowadays.

3. *Traffic Handling:* Like in wired networks, algorithms are mostly concerned with the way of handling communication traffic. For these purposes algorithms give instructions to nodes how to handle a message, without actually changing certain properties, like the power or position, of the node. This leads to the diversity in routing and other algorithms, even when they are designed for the same objectives and network properties. Therefore the way how traffic is handled and how it is routed from sources to destinations, i.e. the type of communication, can be seen as a core aspect of algorithms in wireless networks.

2.3 Type of Communication

Roughly we can distinguish three types of communicating: broadcasting, multicasting and unicasting. We speak of broadcasting if a message has to be sent from one source node to all other nodes in the network. For multicasting several nodes are destination nodes, but not all and not only one. In the case where there is only one source and one destination we speak of unicasting. Usually multicasting and broadcasting are taken together as broadcasting is a special case of multicasting. Of course, unicasting is also special case of multicast, but there are fundamentally differences between multicasting and unicasting that have to be taken into account when an algorithm is designed ([22]):

1. Because of the broadcast nature of the wireless medium using omnidirectional antennas, a message can be received for 'free' within the transmission range. This property is called the *Wireless Multicast Advantage* [55], which provides an advantage mostly for multicasting instead of unicasting. However, the Wireless Multicast Advantage adds also complexity to broadcasting in the network, for example the increased possibility of interference. For a more detailed description of broadcasting problems in MANETs we refer the reader to [47].

2. Sending messages from one node to only one other node can be done in an energy-efficient way by selecting only nodes with sufficient residual energy as relaying node. Multicasting requires that more nodes will be involved and broadcasting requires every node to be a relay node or a receiver. Hence, in the latter case an algorithm has to be developed to reach the nodes with low energy level in an efficient way, e.g. by exploring the Wireless Multicast Advantage.

3. The analysis of routing algorithms is far more easy for unicast routing than for multicast routing, since the flow conservation property, i.e. the sum of the messages on the incoming paths is equal to the sum of the messages on the outgoing paths, holds for all nodes on the path from one source to one destination, but not for all nodes on the paths from one source to multiple destinations. In the multicast routing tree there is at least one node u with at least three neighbors v_1, v_2 and v_3, such that u receives a message from v_1 and relays the message to v_2 and v_3. So, the sum of the messages on the incoming paths is one, only the message coming from v_1, but the some of the messages on the outgoing paths is two, since one message is received at v_2 and one at v_3. Therefore, the conservation property does not hold and instead of using simple and elegant flow formulations, more complex mathematical tools have to be used to analyze the algorithms. We note that this difference is more related to the design of an algorithm than to the actual working of the algorithm, as in unicast routing and multicast routing a node only has to transmit the message once, independent of the number of neighbors.

To characterize the way of communication further, we can distinguish communication algorithms by looking at the number of source nodes, just like we looked at the number of destinations. For interference but also for simulation purposes it is useful to know if multiple broadcasts at the same time can be busy to spread packets over the network. We define the broadcast period of a message as the period describing the time from the moment that a source broadcast a message till the time that all destinations receive the message. Now, three possibilities can be distinguished. The first possibility is that at every moment in time at most one broadcast period is active. In other words, a new broadcast message is sent only when the previous has reached all destinations. Next, we can define the situation in which there are moments in time where several broadcast periods are active, but not of all nodes. There are times in which the network is busy to deliver multiple broadcasts. The last option is that broadcast periods of each node in the network are active at the same time. This situation describes a network in which all nodes send continuously messages, even when the previous message did not arrive yet at all destinations.

We explained the different ways of communication by taking broadcasting as example, but these classes exist also for unicasting and multicasting.

2.4 Network Objectives

In Section 2.1.1 we discussed several types of wireless networks and corresponding tasks. These tasks can be deduced further to objectives for algorithms. The

objectives give an image of how the ideal algorithms should work. The main possible objectives, summarized by Wang and Xiao, are depicted Figure 2.4. Several objectives are strongly related to each other, sometimes positive, sometimes negative. When designing a network it is therefore important to understand the objectives and to define the desired order of importance. In this section we first clarify some aspects of the schematic figure of objectives. Next, we discuss topological objectives, which are only mentioned indirect in the scheme of Wang and Xiao. These objectives are interesting for algorithms that can vary the transmission power of the nodes in order to control the topology.

Maximizing Network Lifetime Due to the common assumption that nodes in MANETs have limited battery capacity, one often used objective is to maximize the network lifetime. However this objective is not always clear, as there are more definitions for network lifetime. Mostly network lifetime is defined to be the time until the first node runs out of energy. Other definitions are the time until a certain percentage of the nodes are without any battery residual capacity or the time until a subsequent number of messages are not delivered to the destination nodes.

Scalability It is undesirable to have computational costs that increase linearly or even higher with the number of nodes in a network. So, we want to have an efficient algorithm that stays also efficient when the network becomes large. Mostly this leads to local algorithms, which we discuss further in Section 2.5.

Robustness In reality unexpected failures can occur. If the objective is to have a *robust* algorithm, one wants a mechanism that can handle such unexpected failures. The algorithm has to fulfill his designated task despite these failures.

Simplicity A reasonable wish is to have wireless sensor nodes that are small and cheap. Due to these wishes one has to assume limited capability and therefore an algorithm has to be simple. Complex computations which demand a lot of memory are not suitable for such nodes.

Sensing Coverage In wireless networks with a monitoring task, it is needed to cover the area that has to be monitored by the deployed sensors. To satisfy this objective, the algorithm has to control the position or sensing area of the nodes.

Network Connectivity The degree of connectivity determines somehow the risk of failures. If there are several distinct paths between a source and a receiver, a message is more likely to be received when one link breaks down compared to the case where there is only one path from source to receiver. A related objective is the guarantee that a network is k-connected. Mostly the powers of the nodes are changed to satisfy this objective as much as possible. Besides connectivity there are other topological objectives for wireless networks that are often desired, like sparseness, symmetry et cetera. We discuss these in Section 2.4.1.

Data Delivery Ratio This ratio is defined as the number of messages that arrived successfully at their destination divided by the total number of messages sent. Robustness is strongly linked to this objective.

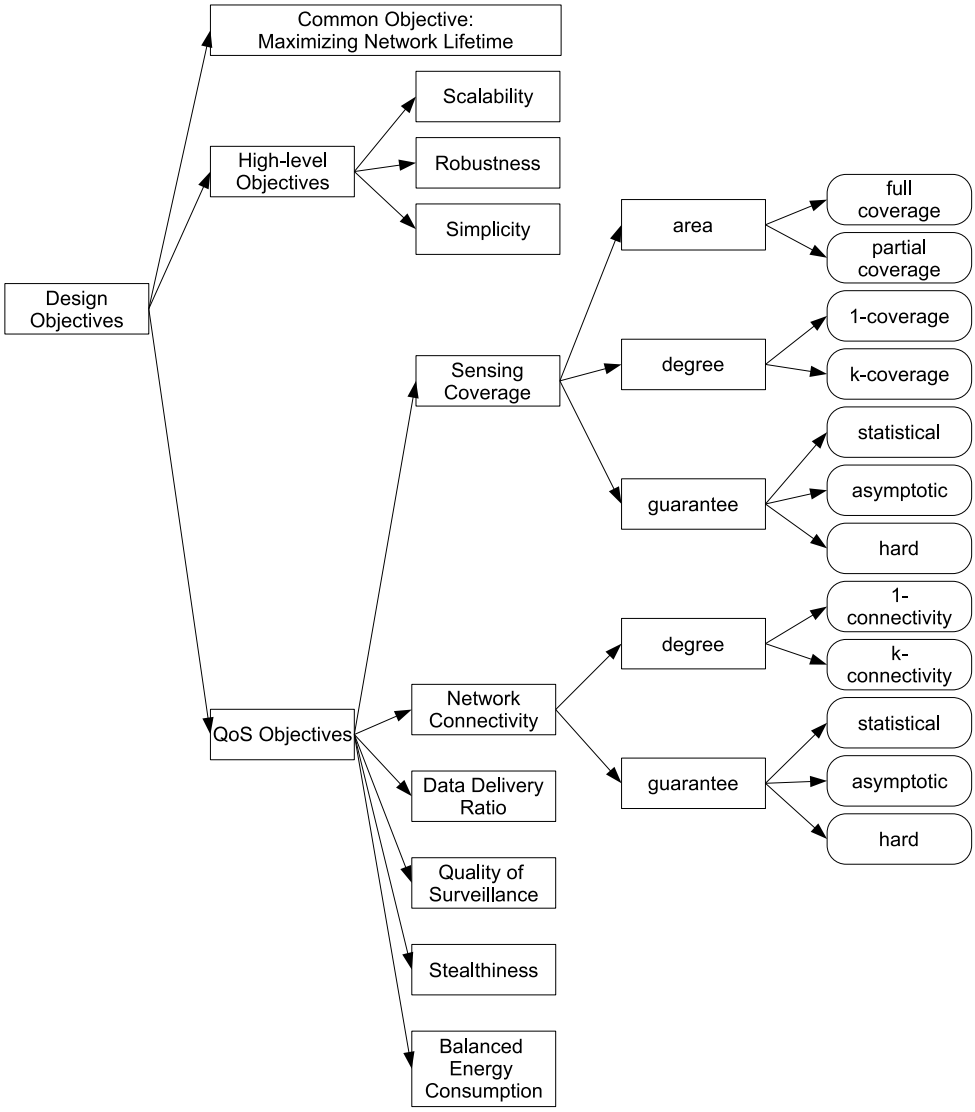


Figure 2.4: Design objectives of scheduling mechanisms (taken over from [50]).

Quality of Surveillance The time until a movable object is detected by a surveillance network depends on the sensing coverage, but also on the deployment of the wireless nodes. This objective is mainly interesting for special monitoring tasks and not for general MANETs.

Stealthiness Military applications of wireless networks require mostly the property to be hard to be detected by the enemy. Short transmitting time and a small number of overhead messages are factors that determine the stealthiness of an algorithm.

Balanced Energy Consumption Depending on the definition of network lifetime, this objective can be seen as a part of the common objective to maximize the network lifetime. To avoid holes in the network by nodes that have no energy supply anymore it is needed that the energy supply is balanced: the goal is to let each node have the same residual energy. In the ideal situation this leads to the maximal network lifetime, as all nodes will run out of battery at the same moment.

2.4.1 Topological Objectives

In wireless networks there are certain topological properties desired to provide a good environment for communication. Topology Control is a technique used in wireless ad hoc networks to find subgraphs of the maxpower communication graph, i.e. the graph in which nodes are assigned maximum power, such that several properties are satisfied. A subgraph can be realized by adjusting the transmission power of the nodes. In general at least the following three properties are desired for a network topology, of which the first property already is mentioned in the scheme of Wang and Xiao:

1. *Connectivity:* If a node is reachable in the network with maximal power, it has to be reachable in the network with adjusted power. This means that the power has to be set in such way that the resulting network is connected. A stronger demand is k-edge connectivity. In such topologies we know that if a link (edge) is not working correctly, there are still other links that can be used to reach a node.

2. *Symmetry:* The symmetry property assures that if a message can be sent to a node, a responding message can also be sent. This property assures us that acknowledgment messages can be used to inform a node of a correct message reception. An algorithm has to take care of the power settings to provide the symmetry property.

3. *Sparseness:* In a sparse graph there are not too many links, formally: $|E| = O(|V|)$. This property reflects that a node should have not too many neighbors in order to reduce interference, which also can be done by adjusting the transmission powers.

Although these three properties already seem to give topologies that are suitable for usage in wireless networking, two additional properties are sometimes wished for, as in [53]. The first one improves the idea of connectivity, the second one of sparseness.

4. *Spanner:* The formal definition is given by the following. Let $G_{max}(V, E)$ be the graph in which every node $u \in V$ send with maximum transmission power. Then the subgraph $\overline{G} \subset G_{max}$ is called a spanner of G_{max} if for every pair u, v of nodes of V the costs $\overline{c}_{uv}$ of a shortest path form u to v in $\overline{G}$ is bounded by a linear function f of the costs c_{uv} of a shortest path from u to v in G_{max}, i.e. $\overline{c}_{uv} \leq f(c_{uv}) \quad \forall u, v \in V$. The costs can be defined in many ways, like for example hop-distances or energy needed to transmit from u to v.

   To illustrate this property, we look at a Minimum Spanning Tree (MST). A MST satisfies the symmetry, connectivity and sparseness properties, but is often not considered as a good topology for communication between two nodes. This is because in the MST topology there may be two nodes that are far away from each other, while they are close to each other or even neighbors in the graph G_{max}. In short, it does not satisfy the spanner property for a certain cost function, as this property restricts the difference in energy costs in the original graph and the resulting topology graph for unicast messages.

5. *Low Degree:* A sparse graph can still have some nodes with many neighbors, look for example at a star graph. To reduce interference also at these nodes, the property *low degree* is introduced. The number of neighbors of a node in the created topology should be bounded from above by a constant, so the maximum degree of all nodes in the network should be less than a constant.

These topological aspects seem to be useful as objectives for controlling a wireless network topology. However, we note that not all mentioned aspects can be satisfied in the same amount. For example, the sparseness objective aims to minimize the number of edges and therefore is opposite to the connectivity objective which needs edges to get a connected graph. Consequently, one needs to value each property in order to know the focus of the algorithm that has to control a topology.

2.5 Algorithm Mechanisms

The last classifier for algorithms we discuss in this thesis, is the type of how an algorithm works in order to satisfy several objectives. We can distinguish centralized, distributed and cluster-based algorithms. Each mechanism has its own specific benefits and drawbacks. Most algorithms in the wireless network area work in a distributed way.

1. *Centralized:* A centralized or global algorithm computes a global solution and presumes the specific node information, like residual energy and position, from each node to be available. A major drawback is that it is not realistic to assume to have all the information available in a central location. Nevertheless, global algorithms are easier to design and give better results, because of the complete information.

2. *Distributed:* For a distributed algorithm the node information is not globally known, but has to be distributed. This simple definition makes it easy

to turn every (global) algorithm into a distributed algorithm by collecting the distributed states, computing the global solution and distributing the solution to the nodes again [51]. A disadvantage of this approach is the high energy consumption resulting from the large amount of communication. Therefore, algorithms that need less communication are requested, which is essentially the idea behind a localized algorithm. A *k-localized* algorithm for some parameter k is a special case of a distributed algorithm, that allows each node to communicate only k times with its neighbors. Each node is also allowed to delay his message. Consequently, a localized algorithm can be very slow. A k-localized algorithm in which a node is not allowed to delay a message is called a *k-local* algorithm. Precise definitions can be found in [51]. In an ad hoc network it is desirable to have local algorithms in order to minimize the energy consumption.

3. *Cluster-based:* In Section 2.1.2 we called a cluster-based network an example of a hierarchical network structure. However, the cluster-based property can also be seen as a mechanism, as it is an intermediate form of a centralized and a distributed algorithm. Specific cluster-heads retreive all available information of the neighbors in its cluster and compute a local solution for the cluster. This can be seen as a centralized algorithm within the cluster. The solutions are then distributed to the other cluster-head to get a global solution. This is typical for a distributed algorithm. Although the cluster-based method has the benefits of both mechanisms (good performance and less overhead compared to a fully centralized algorithm), it also has the drawbacks of these methods (still a lot of overhead costs and special clusterheads consume more energy).

An algorithm can even further be specified by distinguishing online and off-line algorithms. An algorithm that processes information piece by piece without having all the input available from the start is called an *online* algorithm. In contrast to this, *off-line* algorithms assume knowledge of the total input, from the beginning to the end, before computing a solution to a problem. Since an off-line algorithm has more input information it will always compute better solutions than a solution computed by an online algorithm.

Chapter 3

Network Lifetime Problems

In the last chapter we gave a widespread view on classification and design choices for an algorithm for unicasting, multicasting or broadcasting information in a wireless network. In this section we focus on the problem with the objective to maximize the network lifetime, i.e. the Network Lifetime Problem (NLP), and describe four problem categories to classify the problems that correspond with this objective. These problem categories assume a stationary Ad Hoc Network. Therefore this section can be seen as a further classification for a smaller group of algorithms that aim at solving a problem concentrating on the network lifetime in Ad Hoc Networks.

The characterization of the problems is useful to get an overview of the problems already studied and the developed algorithms. It also presents the connections between problems, which can be used to see which parts of a solution has to be changed to compute solutions to another problem category too.

In the first part of this chapter we mention two maximization problems in wireless networks to sketch the context. Next, we focus on maximizing the network lifetime and define four categories of problems that are mostly useful for stationary networks. In Section 3.2.3 we describe the categories in more detail. The final section of this chapter is an example in which the difference between two of the categories is made clear.

3.1 Maximization Problems in Wireless Networks

The main restriction in MANETs is the limited battery capacities, which leads to a wide variety of research problems related to energy-efficiency. Among these problems, there are many maximization problems to improve the performance of a wireless network. Two main maximization objectives occur in literature, which we discuss here.

- *Maximize the Capacity:* The objective is to maximize the capacity of the network, i.e. the number of messages sent successfully through a network. Of course, this type of problem can be further specified by extra assumptions, constraints and objectives. There is for example a difference in the approach of the problem for unicasting or broadcasting. Since we do not focus on this problem, we refer the reader to [23], [32] and [33].

- *Maximize the Network Lifetime:* In Section 2.4 we already listed the interpretations for network lifetime, but we mention them again for clearness. Mostly network lifetime is defined to be the time until the first node runs out of energy, but other definitions are the time until a certain percentage of the nodes are without any battery capacity or the time until a subsequent number of messages are not delivered correctly. We recall that we use the first definition of Network Lifetime. In Chapter 4 we present some literature on algorithms with the objective to maximize the network lifetime.

3.2 Schematic Classification of Network Lifetime Problems

3.2.1 Definitions

To categorize the Network Lifetime Problems we use several terms from the network area. We mention them shortly.

Layered Network In a layered network, the network layers are separated, i.e. protocols in one layer do not interfere with protocols in another layer. For example, the routing protocol, which exists in the routing layer, is not allowed to control the power assignment, which is in the physical layer.

Cross-layered Network Protocols in a specific layer of the network may interfere with other layers in a cross-layered network. So, for example the routing protocol is allowed to change the power assignments of the nodes.

Physical neighbors A node with a certain power assignment has a transmission area. All nodes that are located within the transmission area of the specific node can communicate with that node and are called physical neighbors of that node.

Logical Neighbors The power assignments of nodes lead to a graph structure in which nodes have physical neighbors. Due to computational costs it is often convenient for routing algorithms to reduce this graph to a connected subgraph containing the same nodes but less edges. Note that the physical graph does not change, but only the graph that is used for computational goals. Let us call this graph the logical graph. Then, the neighbors of a node in the logical graph are called logical neighbors. Obviously, all logical neighbors of a node are also physical neighbors of the node. Furthermore, a node can have more physical neighbors than logical neighbors.

Broadcast Tree In a broadcast tree, the root is equal to the source node, the non-leave nodes are equal to the relaying nodes and the leaves are equal to the non-relaying nodes. To be a broadcast tree, the set of relaying nodes plus the root must form a Connected Dominating Set (see Section 4.3.1 for a detailed description of a Connected Dominating Set).

3.2.2 Layered versus Cross-Layered Networks

One of the possibilities to specify the NLP further is by giving the parameters of a network which are considered to be variable. In this section we look at two aspects: transmission power assignment and routing structure, i.e. the structure of the paths from sources to destinations. This characterization can therefore be seen as a combination of two of the defined variables in Section 2.2 with one of the defined objectives in Section 2.4. With respect to these variables there is a big difference between a layered and a cross-layered network. In the layered network the variables can be seen more or less as independent variables, which we can combine. However, the cross-layered network combines the variables into one variable. The routing structure influences the power assignment and vice versa. The problem in cross-layered network is solved by an integral solution in contrary to the solution of the NLP in layered network which exists of a combination of a routing strategy and a power assignment protocol. Because of this difference we discuss these two types of problems separately.

3.2.3 Layered Networks

The transmission power assignment as well as the routing structure can be static or dynamic, that is fixed or variable over a time period. The meaning of a static or dynamic power assignment is clear, but the definition of static and dynamic routing structures needs more explanation. In [22] a routing structure is called static if the structure is not self-reconfigurable. That means for unicasting that for every source-destination pair only the initially chosen route is used for transmitting a message from source to destination. In case of broadcasting, all broadcast messages from a source node are spread over the network using the same broadcast tree. Note that different source nodes may have different broadcast trees, but that only one broadcast tree per source node may be used during a given period. For multicasting a similar definition holds. Obviously, for mobile networks there is never a guarantee that the initial routing structure is sufficient for delivering all messages during time, as node can move to other positions. Therefore, this classification is only useful for NLPs in stationary Ad Hoc Networks, which we also study in Part II of this thesis.

We assume in this thesis that the routing structure is determined given the power assignment, otherwise one can not even speak properly about a network. Adapting the definition in [22] to this case, we call a routing structure in a network with dynamic power assignment static, if the structure does not change during the period that a specific power assignment holds. If the power assignment changes in a static routing structure situation, a new period starts and, given the new power assignment, a new routing structure is allowed.

By combining the two possibilities of each variable we can define four categories of the NLP in stationary Ad Hoc Networks, which we list below.

1. Static Power Assignment and Static Routing Structure
2. Static Power Assignment and Dynamic Routing Structure
3. Dynamic Power Assignment and Static Routing Structure
4. Dynamic Power Assignment and Dynamic Routing Structure

In Table 3.1 a scheme is presented that shows the connections between the resulting problems and gives a short characterization of each problem. The four problems summarized in the scheme are discussed in the next section. One general observation is obviously, that the more variables can be adjusted dynamically, the longer the network may live. Mainly, because of the possibility to react on the residual energies of the batteries. By this, direct action can be taken to save nodes that have low energy by choosing other nodes as relay nodes.

		Routing Structure	
		Static	*Dynamic*
Power Assignment	*Static*	SPSR • Physical Neighbors are fixed • Logical Neighbors are fixed	SPDR • Physical Neighbors are fixed • Logical Neighbors are variable
	Dynamic	DPSR • Physical Neighbors are variable • Logical Neighbors are fixed given a power assignment	DPDR • Physical Neighbors are variable • Logical Neighbors are variable given a power assignment

Table 3.1: Scheme of four categories of the NLP in a wireless, stationary and layered Ad Hoc Network.

Static Power Assignment and Static Routing Structure

Abbreviation: SPSR.

Situation: In the case where both the transmit power and routing structure are static, the initial power assignment (e.g. maximum power) and initial routing structure do not vary over time. So, the physical and logical neighbors stay the same. Therefore the initial choices for power assignment and the routing algorithm that produces a routing structure determine fully the behavior of the network.

Problem: Since the initial choice for power assignment and routing structure is crucial for this situation, the goal is to find a power assignment and routing structure that is robust and that maximizes the network lifetime subjected to some additional constraints.

In Ad Hoc Networks it is often assumed that initially the nodes are not aware of other nodes. Therefore, normally an initialization procedure is processed in which the nodes transmit a message with maximal power to

announce themselves. This packet can also contain other information, like location and battery status, which can be used to compute a topology and corresponding power assignment that satisfies all requested constraints in an efficient way. However, if an initialization is not possible or allowed, a power assignment has to be selected, without any knowledge of the network. If network connectivity is one of the constraints, then assigning each node its maximal power gives the most guarantee for a connected network, but may be not the best power assignment.

Choosing a routing strategy that computes routing structures that are energy efficient without knowledge of the type of communication and the sendpattern is difficult. If we specifically look at broadcasting with only one source node that stays the same and assume that the battery capacities are known, the best objective is to find a broadcast tree such that the nodes with the lowest capacities become leaves. Nodes with much energy left should become relay nodes. Minimum Spanning Tree algorithms are often useful to compute such a broadcast tree.

Static Power Assignment and Dynamic Routing Structure

ABBREVIATION: SPDR.

SITUATION: In this situation the initial power assignment does not vary, but the routing structure may. Or, equivalently, the physical neighbors stay the same during time, but the logical neighbors may change. This leads to a collection of routing structures with corresponding time periods in which these structures are used. We call this collection a *routing schedule*.

PROBLEM: Similar to the previous problem, if an initialization is allowed, a power assignment can be computed using the knowledge obtained in the initialization phase. The goal of the power assignment is to give a network with enough variability in producing routing schedules, but to assign not too much power to nodes with low batteries. If there is no initialization phase, a connected network is guaranteed most if all nodes are assigned maximal power.

The variability in producing routing schedules gives possibilities to choose at each moment the routing schedule that maximizes the network lifetime. The schedule does not have to be computed before the initialization of the network, but may also be created during time. Additional constraints and objectives can be chosen from a wide variety that consists of properties like for example sparseness and symmetry. The main difficulty of the problems in this category is to define an objective function that determines the quality of a routing structure at a specific moment in time. This can be used by an algorithm to determine when a new routing structure should be created and how this structure should look like.

Dynamic Power Assignment and Static Routing Structure

ABBREVIATION: DPSR.

SITUATION: Since the power assignment may change during time in this situation, also the physical neighbors of a node may change. Similar to a routing schedule we call the list of power assignments with their corresponding time periods a *power schedule* ([6]). During a time period in the power schedule, the routing structure is static. So, the routing structure is determined given the power assignment and consequently a node keeps the same logical neighbors during a time period.

PROBLEM: The dynamic power assignment makes an initialization phase possible. To announce themselves nodes are assigned first maximal power. Next, when network information is available, a more appropriate power assignment can be selected to reduce energy consumption. The goal of power assignment protocols is to select the power assignment satisfying several network constraints, while maximizing the network lifetime. These problems are also called Topology Control problems.

Similar to the problem description of SPSR, the task of the routing protocol is to find a routing structure that maximizes the network lifetime by selecting nodes that have to relay and nodes that only receive the message, the so called leaves in the broadcast tree. The algorithms should have criteria for selecting logical neighbors in order to maximize the network lifetime.

Dynamic Power Assignment and Dynamic Routing Structure

ABBREVIATION: DPDR.

SITUATION: Both the transmit power and the routing structure may be changed in this situation and consequently also the physical and logical neighbors of a node, which leads to more and better options to maximize the network lifetime. The dynamical adjustments make it possible to adapt to situations with specific locations and residual battery capacities.

PROBLEM: As for all the other situations, the objective is still to maximize the network lifetime subjected to constraints (e.g. sparseness, symmetry et cetera). Since a common assumption in a layered network situation is that it is not possible for the protocol to change a power assignment and routing structure at the same moment, the problems can be treated as a combination of the problems DPSR and SPDR. For example, assume that we want to change the routing structure in the network. A natural way to solve this problem, is to assume the transmission temporarily to be fixed and the routing structure to be dynamic. This is the equivalent of the SPDR problem in which we select a routing structure with the objective to maximize the network lifetime using certain protocols. The next step is to investigate if we can lower some powers. Therefore we fix the routing structure temporarily and let the transmit power to be dynamic, which is essentially a DPSR problem. Solving this, we dynamically change the routing structure and transmit power. One also can do it the other way around by changing first the transmit power with fixed routing structure (DPSR problem) and then the routing structure with fixed power assignment (SPDR problem).

3.2.4 Cross-layered Networks

As mentioned earlier, the cross-layered network has an integral approach to the routing structure and the transmission power. However, in many cases the focus lies on the routing structure. Based on the routing structures the powers are minimized as much as possible. More specifically, if a route is selected from source to destination, the powers of the nodes in the network are adjusted to the minimal power while keeping the route alive. Just as in layered network, we can distinguish a dynamic and a static case for a cross-layered network. But now we do get two classes instead of four, as the power and routing structure are taken as one variable. The static variant in which only one routing structure in combination with a power assignment is allowed over time is called the Static Network Lifetime Problem and the dynamic variant in which multiple routing structures with corresponding power assignments are allowed is called the Dynamic Network Lifetime Problem.

Static Network Lifetime Problem

Abbreviation: SNLP.

Situation: The routing structure does not vary over time and because of the integral approach the power assignment is also static. Equivalently, both the physical and logical neighbors of a node stay the same over time. This problem resembles therefore more or less the SPSR problem.

Problem: Similar to the SPSR problem one would like to find a broadcast tree in which nodes with low energy are leaves and nodes with high energy have to relay the messages. Kang and Poovendran discuss this problem in [22] and present an adaptation of Prim's algorithm to compute the Maximum Static Network Lifetime. They also prove for the special case in which every node has the same amount of initial energy (a so called *Equally Distributed Energy Network (EDEN)*) that the globally optimal solution to the static network lifetime maximization problem is a Minimum Spanning Tree (MST). Note that the MST solution is not unique in general.

Dynamic Network Lifetime Problem

Abbreviation: DNLP.

Situation: Multiple routing structures may be used for communication in this situation and the power assignments may be adjusted to use as less power as possible for a routing structure. So, the physical and logical neighbors may change continuously in this situation. If battery information is available it is also possible, just like in the DPDR problem, to use the battery information to enhance the routing protocol. Because of these two reasons, the network lifetime in the DNLP is longer than in the SNLP. This is discussed further in Section 3.2.4.

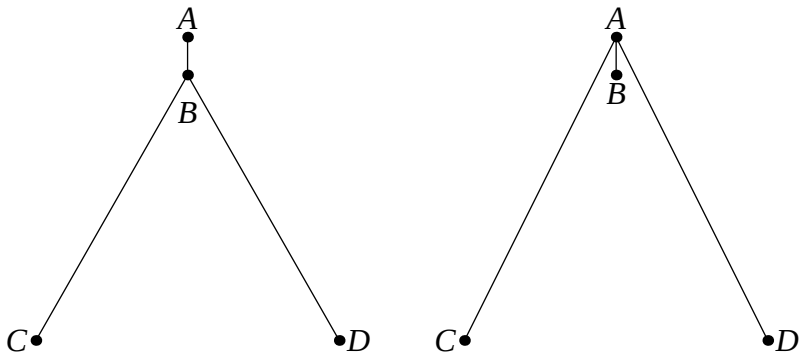
Problem: Even though the power assignment and the routing structures are also dynamic in the DPDR problem, the DNLP problem is much different from DPDR. The cross-layered network has an integral approach and can compute both the power assignments and routing structure at

the same time, in contrary to the DPDR situation in a layered network in which the problem has to be splitted into two subproblems. This integral approach needs therefore integral algorithms and no combination of topology control algorithms and routing protocols if one wants to use the strength of a cross-layered network.

Example of Maximal Network Lifetime in SNLP and DNLP

In the remaining of this subsection we look at broadcasting a message over a network with one source node using a broadcast tree and study the difference between the maximal network lifetime in the SNLP and the DNLP. We do not look into methods to create the broadcast tree. In the SNLP only one broadcast tree is chosen and it will not be updated, while in the dynamic problem several trees may be used to broadcast messages over the network. We demonstrate with a simple example, as shown in Figure 3.1, that the dynamic variant has a greater or equal maximum network lifetime compared to the static variant.

Initially, assuming that the transmission power attenuates quadratically or more and all nodes have the same initial battery supply, the broadcast tree that maximizes the network lifetime when node A wants to broadcast a message is the tree shown in Fig. 3.1(a). Sending from A to C and D is more expensive than sending it from A via B to C and D. Therefore, the broadcast tree in Fig. 3.1(a) would initially be chosen as the routing structure for the optimal strategy. It is easy to see that the DNLP achieves a longer maximal network lifetime. Let us use the optimal broadcast tree from the the SNLP until node B runs out of energy. If we are allowed to update our broadcast tree by changing it into the tree as in Figure 3.1(b), we can consume the residual battery energy that is left for node A. We know for sure that there is some residual energy as both nodes (A and B) started with the same initial supply and node B needed to cover a bigger transmission area and used more transmission power than A. If we assume nodes A and B to have almost the same position, we can state that the maximal network lifetime in the DNLP is twice the maximal network lifetime in the SNLP. This example indicates that the maximal network lifetime for the DNLP is always bigger or equal than the maximal network lifetime for SNLP. The strategy is to use the optimal solution of the SNLP and look for another broadcast tree to be created to extend the network lifetime in the DNLP.



(a) Optimal broadcast tree for SNLP. (b) Other possible broadcast tree.

Figure 3.1: Two possible broadcast trees for broadcasting the network from node A.

Chapter 4

Algorithms for the Network Lifetime Problem

In the last seven years many algorithms have been developed addressing the Network Lifetime Problem. As mentioned in the first chapter, most algorithms are designed for a specific network and a specific task. In this chapter we focus on algorithms that address the topology control problem and routing structures for broadcasting. Some of these algorithms assume cross-layered networks, but also some assume layered networks. We do not intend to give a complete overview, but present some algorithms to get a basic idea of the existing work in this research area.

Generally, algorithms that have the objective to maximize the network lifetime work with less complex objectives. In literature there are two common used classes of objectives, which we discuss in Section 4.1. The next section addresses the problem of the balance between a sophisticated algorithm and the overhead costs in terms of flooding versus topology control. After this section, the remaining of the chapter discusses existing algorithms in the field of topology control and broadcasting in wireless networks with the objective to maximize the network lifetime.

4.1 Algorithm Objectives

In literature, there are in general two classes of approaches to find an objective function that defines the quality of a routing structure. The first class is to minimize the total costs or to maximize the total profits ([3], [5], [8], [13], [31], [21], [22], [49], [55], [42]), the second is to minimize the maximal costs of the nodes ([22]) or to maximize the minimal profit of all nodes ([11], [18]). We denote the first objectives respectively by MinTotal and MaxTotal and the second objectives by respectively MinMax and MaxMin. The cost and profits can be defined in many ways, like the hop distance, the power needed to transmit, the power already used to transmit and a function that combines residual energy and transmitting power et cetera. The cost or profit definition combined with one of the objectives determines the choice for a routing structure and routing schedule. We discuss the approaches by looking at two cases. First we study the MinTotal and MinMax approach with respect to the transmission

power. Next, we study the MaxTotal and MaxMin approach with respect to the residual energy. After both studies we present our conclusion concerning these approaches.

In this section we assume a cross-layered network and nodes that only need energy for transmission and not for reception. A node with zero energy can therefore still function in the network as long as it does not have to relay a message. We also assume that the time between two broadcast messages is fixed and such that during this time the message is propagated to the complete network. Therefore, the network lifetime can be expressed in the number of broadcast messages that arrived at all destination nodes.

4.1.1 MinTotal versus MinMax Transmission Power

A network topology is defined by the nodes together with their locations and the links between the nodes. Links are established by assigning sufficient transmission power to the nodes at both sides of the link in order to be able to transmit from one node to the other. We discuss the problem of power assignment in order to maximize the network lifetime by looking at the two approaches MinMax and MinTotal.

Kang and Poovendran study the differences between the MinTotal and MinMax approach with respect to the transmission power in [22] and mention (in 2005) that many research papers focus on MinTotal criteria, even though it is not sure whether the heuristics give good solutions. In their paper they compare several existing algorithms and conclude from simulations that the MinMax optimization for power assignment provides better or comparable results than the MinTotal strategy in terms of the network lifetime. Their conclusion may be true when taking a number of random networks, but in this section we show that the performance of an approach depends also on the sendpattern. We show this by exploring the two different approaches on specific networks, but first present some known results on this topic.

Literature

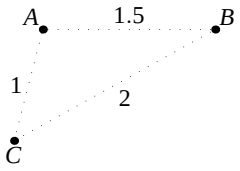
Liang [31] and Cagalj et al. [5] have proven independently that the minimum-energy broadcast problem with the objective of minimizing the total transmitted power is NP-hard. The statement is even stronger, Cagalj et al. have shown that it is unlikely to have an approximation algorithm ratio better than $\Omega(\log n)$ unless $P = NP$. The weighted graph version of the minimum-energy broadcast problem is also NP-hard in metric space, when the transmissions are restricted to a given set of power levels by means of an upperbound on the allowed radius [15].

During our research we did not see any results regarding minimizing the maximum energy consumption. However, there is a centralized algorithm that solves the problem in polynomial time. The procedure is to remove the links in descending order with respect to the transmission power needed for the link and continue as long as the network stays connected. If no links can be removed anymore, the resulting graph is the solution to the problem to find the topology that minimizes the maximum assigned power in the network.

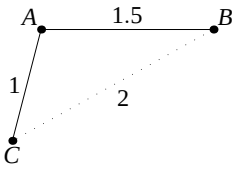
There are various heuristic algorithms based on the MinTotal and MinMax approaches exist. We study the approaches in depth in the following sections.

MinMax

The MinMax approach with respect to the transmission power tries to get a connected topology with the lowest maximal transmission power needed for a node to broadcast a message over the network. The topology can be created by using the procedure described above. As a consequence of the procedure, the resulting topology is independent of the node that want to broadcast a message, or, stated differently, there is only one broadcast tree used in the network independent of the source node. In Figure 4.1 we present an example network and the transmission powers needed to establish a link. Clearly, the link between nodes B and C demands the maximum transmission power. Therefore the optimal topology using this optimization is a network in which there is no link needed between B and C to get a connected network. The resulting network is shown in Figure 4.1(b).



(a) The original network with on the edges the transmission power needed to communicate between two neighbors.



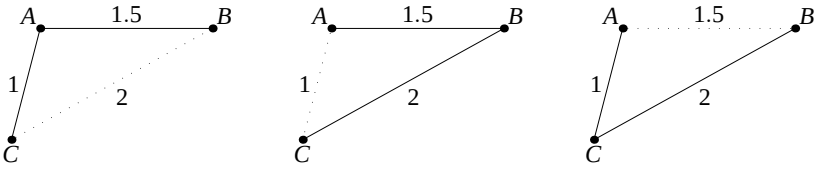
(b) The MinMax approach from the viewpoint of nodes A, B and C.

Figure 4.1: The MinMax approach applied to the transmission powers in a triangular network.

MinTotal

As mentioned earlier, the Wireless Multicast Advantage is that in wireless transmissions every node that is in the tranmission area receives the transmitted data. This advantage is important for minimizing the total transmission power. As each node has different neighbors, the advantages are not equal for each node. If we want to look at the MinTotal approach we therefore also need to look at the approach from a certain viewpoint, the MinTotal with respect to for example node A. So, in contrast to the MinMax approach, the MinTotal approach depends on the node that wants to broadcast or in other terminology, the broadcast tree can differ per source node. The MinTotal goal is to broadcast a message from a node to all nodes in the network while using a minimum total transmission power. To compute the MinTotal costs we sum the transmission powers of the nodes that are used for transmission from source to destinations. So, we look only at the transmission power needed to get a unidirectional broadcast tree with the source node as the root. In Figure 4.2 all the MinTotal networks of our example are given. We explain the figure by discussing Figure 4.2(a). In the resulting network node A has a link to B and C. For the link $A-B$ a power of 1.5 is needed, which is also sufficient to establish the link $A-C$, so the total power costs are 1.5. The other option is to use C as a multihop node to reach B. However, the total power costs are in this option 3.5, the sum of the needed

powers for A and C. Also for the other nodes as sources, the MinTotal approach does not use multihop since the additional costs to reach an extra neighbor are less than using a node as a multihop. This does not hold in general, because it is realistic to assume the transmit power to be at least quadratically related to the distance, which makes the multihop approach cheaper in many situations.



(a) The MinTotal approach when node A sends a packet.

(b) The MinTotal approach when node B sends a packet.

(c) The MinTotal approach when node C sends a packet.

Figure 4.2: The MinTotal approach applied to the transmission powers in a triangular network.

MinMax versus MinTotal

MinMax can outperform MinTotal and vice versa in terms of Network Lifetime, depending on the initial battery capacities and a sendpattern. Assume for example that the nodes in the depicted network have initial battery capacities $A = 5$, $B = 5$ and $C = 5$. We now give two different (broadcast) sendpatterns and show that for one sendpattern the MinTotal approach is better for the network lifetime and for the other sendpattern the MinMax approach is better for the network lifetime. In Table 4.1 the processing of sendpattern $A - A - A - B$ is depicted and in Table 4.2 the processing of $B - B - B$. $E(A)$ and $E(B)$ denote respectively the energy of node A and the energy of node B. The energy of node C does not change and is therefore not listed in the tables. We assume in these examples that the energy costs for nodes to send a packet are equal to the transmission powers of the nodes. In each column of the tables one can find the resulting energy after a message is broadcasted with the node listed in the first row as source. Note that in the MinTotal approach different topologies are used for different sources. This rather simple example shows clearly that there is no overall winner, but that the performance depends on the sendpattern.

4.1.2 MaxTotal versus MinMax Residual Energy

Optimizing with respect to the transmission power alone does not take into account the energy left at each node and is therefore not able to maximize network lifetime in a smart way. Because of this argument it seems interesting to optimize with respect to the remaining residual energy. In this section we discuss this issue by looking at the MaxMin and the MinTotal approach and find that also for these approaches the performance in terms of network lifetime depends on the sendpattern.

(a) Processing the sendpattern on a network in which the powers are assigned to the nodes using a MinMax approach.

	Initial	A	A	A	B
$E(A)$	5	3.5	2	0.5	-0.5
$E(B)$	5	5	5	5	3.5

(b) Processing the sendpattern on a network in which the powers are assigned to the nodes using a MinTotal approach.

	Initial	A	A	A	B
$E(A)$	5	3.5	2	0.5	0.5
$E(B)$	5	5	5	5	3.5

Table 4.1: Processing a $A - A - A - B$ sendpattern using the MinMax and MinTotal approach for the power assignment.

(a) Processing the sendpattern on a topology created by a MinMax approach.

	Initial	B	B	B
$E(A)$	5	4	3	2
$E(B)$	5	3.5	2	0.5

(b) Processing the sendpattern on a topology created by a MinTotal approach.

	Initial	B	B	B
$E(A)$	5	5	5	5
$E(B)$	5	3	1	-1

Table 4.2: Processing a $B - B - B$ sendpattern on different created topologies.

Literature

The problem to maximize the total residual energy is equal to the problem to minimize the total energy consumption. Therefore the results of Liang [31] and Cagalj et al. [5] as discussed in Section 4.1.1 also hold for this problem.

Like for the problem to minimize the maximum power in a network (cf. Section 4.1.1), there exists also an algorithm that finds a broadcast tree that maximizes the minimum residual battery capacity in polynomial time. The algorithm starts with a set T that contains only the source node. Each round, the neighbor of T that has the maximum residual energy and reaches also nodes that are not in T is added to T. This procedure continues, until all nodes are in T or in the neighborhood of a node in T. In [34] another algorithm can be found that solves the problem in polynomial time. The authors of [34] also proof that the problem of finding a maximum residual battery capacity broadcast tree with minimum total energy consumption is NP-complete.

MaxMin

An interpretation of maximizing the minimum residual energy is the following. One wants to choose relay nodes as much as possible in a way that the bottleneck nodes, i.e. nodes that have relatively little energy left, become leaves in the broadcast structure.

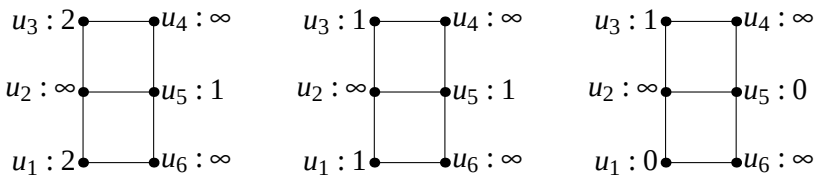
MinTotal

The MinTotal approach with respect to available energy is rather straightforward: choose relay nodes such that as less energy as possible is used to send a message to all the nodes in the network.

MaxMin vs MinTotal

Just like for the optimization with respect to the needed transmission power, there are examples for the optimization with respect to the residual energy in which the best approach depends on the sendpattern. We first examine the case where the MinTotal strategy outperforms the MaxMin approach. In Figure 4.3 an example is given of the processing of the sendpattern $u_2 - u_1 - u_1$ choosing the MaxMin strategy. In the example we assume the energy costs to relay a message to be equal to 1 for each node. The available energy of each node is displayed after the colon. We use ∞ to denote that a node has sufficient energy and therefore its exact energy value is not displayed. In the first broadcast of the sendpattern nodes u_1 and u_2 have to relay the message to save bottleneck node u_5. The second broadcast of the sendpattern, which is the first broadcast initiated at u_1, uses u_2, u_6 and one of the nodes u_3 and u_5. Both can be chosen and it is not important to the example which one is chosen. But since it is more likely that node u_5 is chosen as u_5 has more neighbors, we assume that node u_5 is used for relaying. The third message of the sendpattern can not be sent as there is no power left at u_1. So, using the MaxMin approach only two messages of the sendpattern can be broadcasted. Figure 4.4 shows that the sendpattern $u_2 - u_1 - u_1$ can be processed completely using a MinTotal strategy. For the first broadcast only node u_5 is chosen to reach the whole network. The other two message are initiated at u_1 and relayed by u_2 and u_3.

On the other hand there exist also a simple example in which the MaxMin outperforms the MinTotal strategy: take the same initial conditions and process the sendpattern $u_2 - u_5$. Clearly, by saving the bottleneck node it is possible for node u_5 to send a message in the MaxMin approach, which is not the case using the MinTotal strategy.

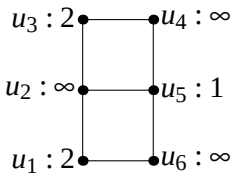


(a) Original network with initial battery values.

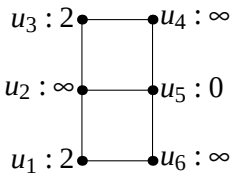
(b) Resulting network after sending the first message of the sendpattern. The message is broadcasted via nodes u_2, u_5 and u_6.

(c) Resulting network after sending the second message of the sendpattern, a message from u_1.

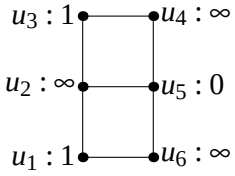
Figure 4.3: Broadcasting sendpattern $u_2 - u_1 - u_1$ using a MaxMin approach with respect to the energy to save total energy.



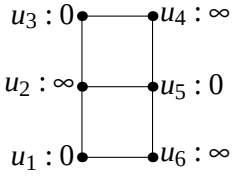
(a) Original network with initial battery values.



(b) Resulting network after sending the first message of the sendpattern, a message from u_2. The message is relayed by node u_5.



(c) Resulting network after sending the second message of the sendpattern, a message from u_1.



(d) Resulting network after sending the third message of the sendpattern, a message from u_1.

Figure 4.4: Broadcasting sendpattern $u_2 - u_1 - u_1$ using a MinTotal approach with respect to the batteries to save total energy.

4.1.3 Conclusion

In this section we observed the MinMax and MinTotal approaches with respect to the needed transmission power and looked at their performances in terms of network lifetime. By examples we showed that there is no optimal strategy for all situations, but that the best approach depends on the sendpattern. Therefore, one has to be a bit careful with the conclusion of Kang and Poovendran in [22] that the MinMax approach outperforms the MinTotal strategy. Their conclusion is mainly based on simulating both approaches in 100 random networks, in contrary to our examples with specifically designed networks.

Similarly to the result for transmission power, the performances of the MaxMin and MinTotal approach for residual battery capacity also depend strongly on the sendpattern. Although to our knowledge there has not yet been an extensive study on the performances of the MaxMin and MinTotal strategy with respect to battery in random networks, we expect in general the MaxMin approach to be better. The main reason for this is that the MaxMin strategy looks at the battery status of the whole network and not only at the energy consumption. Since the network lifetime is defined as the time that the first node runs out of energy, it is necessary to use battery information to save nodes with little energy.

However, an important aspect is that MaxMin generally consumes more energy by selecting other relay nodes in order to save the node with lowest energy value. If this node almost runs out of energy, this is of course the best option, but if all node still have a lot of energy it is doubtful if the increased

energy consumption will pay off later on. For stationary networks it can be expected to be not too bad, as the network does not change and so the energy consumption will be balanced. But for mobile networks it is certainly not sure that consuming more energy to save a node will pay off later on. Mainly because of the fact that now not only the sendpattern is unpredictable, but also the position.

Using a combination of both approaches seems therefore to be an interesting topic of research. Since we do not cover that topic in this thesis, it has to be studied in future work.

4.2 Smart Algorithms versus Low Overhead Algorithms

In general, to create a sophisticated and smart algorithm, much overhead data is necessary to give the algorithm enough information to take good decisions. But a simple algorithm needs less information and has therefore also less overhead. We clarify this issue by looking at the balance of overhead between topology control and flooding with maximal power. We assume that the topology control algorithms have the task to create an energy-efficient topology by reducing powers to get a better structure for routing messages. Flooding with maximal power is the strategy to broadcast messages by instructing every recipient of a message to retransmit the message with maximal power. In Section 4.3.2 we explain flooding further.

If we consider a stationary network, we know that nothing will change about the network structure. In such cases we can use some energy to construct a topology needed for finding an optimal broadcast tree. In the Dynamic Network Lifetime Problem, we will use more broadcast trees and can find them easier and cheaper in the created topology, a subgraph, than in the maximal power topology, the total graph. So we may expect that spending some energy to create a good topology will pay off later on. In case of a high-mobile network, the network changes constantly and there is a little chance that the initial created topology can be used for a longer period. If it can be used (i.e. it is still connected), it is the question whether it is the most optimal solution in the new situation. To create a subgraph topology, messages have to be sent to all the nodes. If we skip this stage, we take more risk, but can save energy. Flooding with maximal power is an expensive solution, but may still be cheaper than running first the topology algorithm and then routing it via a computed broadcast tree as it saves two algorithmic topology computations.

An interesting idea is of course to find an intermediate form. This will be something like a repairing stage in which we repair the original topology. To repair the topology, a new power assignment has to be computed based on the new situation and therefore topology control messages have to be sent. The more links have to be repaired, the more energy is consumed by the network to repair the network. Consequently, the repairing stage is more interesting for a sparse graph, as in a high-mobile network the total graph continuously changes and therefore a lot of links will have to be re-established. The most extreme sparse graph that is still usable for broadcasting is the broadcast tree and therefore it is likely that it will cost less energy to repair this tree. But

it is also likely that we often come to the conclusion that the tree can not be repaired (due to the combination of maximal transmission range and mobility) and a new tree has to be created. Therefore it is doubtful whether such an intermediate form will be more energy-efficient than the two other approaches.

4.3 Algorithms

After having discussed two issues concering algorithms, we now describe some existing algorithms concering the Network Lifetime Problem. Since the problem in a layered network could be splitted into the problems of topology control and routing, we discuss the algorithms in these two classes. The algorithms that assume a non-layered network are presented at the section that discusses the routing algorithms, as those algorithms focus on the routing of a packet and create a topology that suits the routing structure. In this section we still denote with $G(V, E)$ a connected graph with a node set V and an edge set E. However, we now assume G to be the graph in which all nodes are assigned the maximal power. A topology created by an algorithm is therefore a subgraph of G. In the graphs, an edge between the nodes $u, v \in V$ is denoted by uv.

4.3.1 Topology Control

The idea behind topology control is to reduce the number of edges in G, the graph created by sending with maximal power, to get a subgraph $\overline{G}$ that satisfies certain properties. By reducing the number of links and adapting the power to only these links, energy can be saved which makes it mostly possible to prolong the network lifetime. We note that none of the discussed algorithms has specific instructions for the power assignment. However, a simple extension is possible to each algorithm such that all node are assigned the minimum of power while preserving the links to the neighbors defined by the created topology. This clarifies the strong relation between topology control and power assignment. None of these algorithms aim at maximizing the minimum residual energy of the nodes or minimizing the maximum transmit power of the nodes. The main issue is to find a topology with less edges and consequently less transmit power. For the algorithm, it is not that important which edges are removed. The strategy concerning maximizing the network lifetime is therefore more or less a MinTotal approach with respect to the consumed energy.

After finding a topology graph one can choose for a specific routing structure or routing schedule. We discuss shortly some of the geometrical graph structures that are created by topology control algorithms. Since the topological structure is strongly related to a connected dominating set, we also give a definition for this. The first six structure are discussed extensively in [26]. In Figure 4.5 also example topologies of these structures are given. The two algorithms that follow upon these structures are more complex, but give a good view on the work that exists on topology control algorithms in wireless networks.

Connected Dominating Set

A *Dominating Set* is the set of nodes $S \subset V$ such that each node of G is in S or is a neighbor of a node in S. A Dominating Set S is a *Connected Dominating*

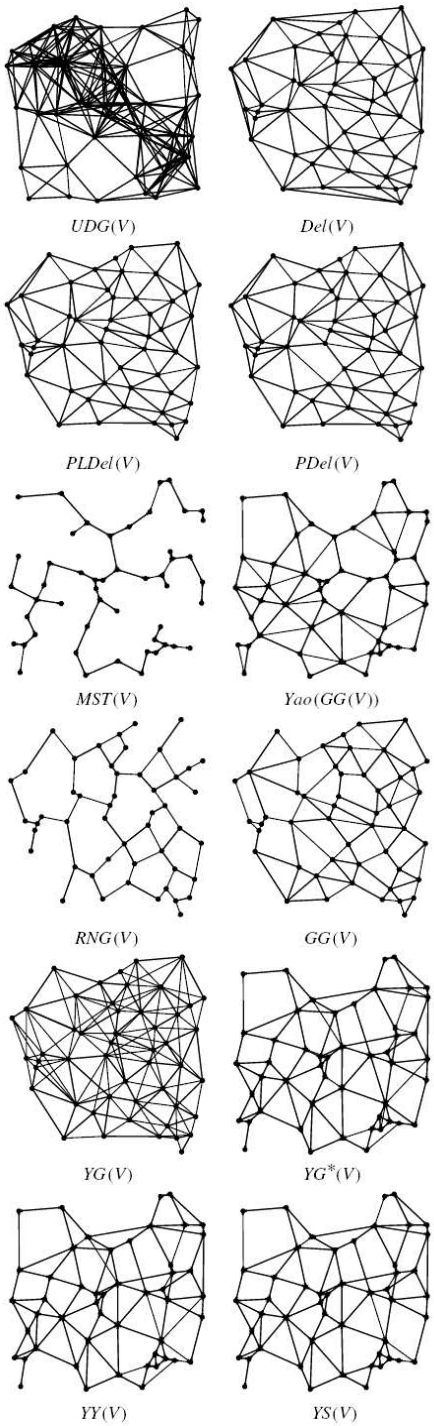


Figure 4.5: Topology examples taken over from [26].

Set (CDS) if S induces a connected subgraph of G. The induced graph of a CDS is a topology.

Minimum Spanning Tree

A Minimum Spanning Tree of G ($MST(G)$) is a tree in G that connects all nodes with minimum total edge length. Many algorithms are based on MST by changing the edge lengths into some cost definition for every edge. However, the principle is still to minimize the total costs.

Relative Neighborhood Graph

The Relative Neighborhood Graph ($RNG(G)$) is defined as follows. An edge uv is included in $RNG(G)$ if in the intersection of the two circles centered at u and v with radius $d_g(u, v)$ there is no other node $w \in V$ such that at least one of the edges uw or vw is in E. So, the lune (the intersection of the circles) may not contain other complete edges that have an endpoint at u or v. See Figure 4.6(a) for an example.

Gabriel Graph

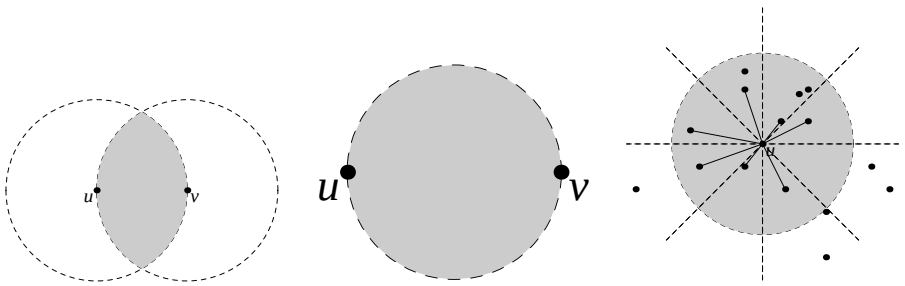
The Gabriel Graph ($GG(G)$) is similar to the RNG but uses an other area instead of the lune. It works with the disk with diameter $d_g(u, v)$ containing both u and v. An edge uv is included in the Gabriel Graph if there are no other edges with endpoints at u or v in this disk. Figure 4.6(b) shows an example.

Yao Graph

The Yao Graph with an integer parameter $k \geq 6$ is denoted by $\overrightarrow{YG}_k(G)$. The algorithm divides the transmission area round a node u in k equally sized cones. In each cone the shortest edge uv among all the edges is chosen and a directed link $\overrightarrow{uv}$ is added. Ties are broken arbitrarily. If this procedure has been applied at all cones, the resultant graph is the Yao Graph. An example is given in Figure 4.6(c). The *reverse* Yao Graph is similar to the Yao Graph, but adds the reversed link $\overrightarrow{vu}$ instead of $\overrightarrow{uv}$. Other modified Yao graphs are Yao-Yao ($YY_k(G)$) [30], Symmetric Yao Graph ($YS_k(G)$) [30] and Sink Structure Yao Graph ($\overrightarrow{YG}^*_k(G)$) [29].

Delaunay Triangulation

Let us assume that there are no four nodes in V that are co-circular, i.e. there exist no circle that goes through four nodes. A triangulation of V is a Delaunay Triangulation ($Del(V)$ or $DT(V)$) if the circumcircle of each of it triangles does not contain any other nodes of V in its interior. There are several modifications of this algorithm, like Unit Delaunay Triangulation ($UDel$ or UDT) [27], Localized Delaunay Triangulation ($LDel$ or LDT) [27], Planar Localized Delaunay Triangulation ($PLDel$ or $PLDT$) [27], Restricted Delaunay Triangulation ($RDel$ or RDT) [17] and Partial Delaunay Triangulation ($PDel$ or PDT) [28].



(a) The Relative Neighborhood Graph: the edge (u, v) exists only when there is no node in the colored lune (excluding its boundary) that is connected with u or v.

(b) The Grabriel Graph: the edge (u, v) exists only when there is no node in the colored circle (including its boundary) that is connected with u and v.

(c) The Yao Graph with $k = 8$: the edge (u, v) exists only when there is no other node with shorter distance in the cone that is bounded by the transmission range.

Figure 4.6: Several examples for topology structures.

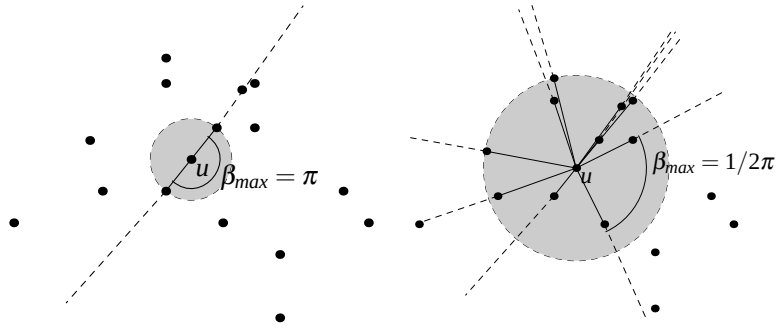
Cone Based Topology Control

The Cone Based Topology Control algorithm (CBTC) [25], [52] constructs a topology in which the neighbor set $\{v_1, v_2, ..., v_k\}$ of a node u has to satisfy the following condition: if a disk centered at u is divided into k cones by lines uv_i $(1 \leq i \leq k)$, then the angle of the maximum cone is no more than α. In [25] it is proven that, when $\alpha \leq 5\pi/6$, connectivity is preserved and when $\alpha \leq 2\pi/3$ the corresponding symmetric subgraph (after removing unidirectional edges) is connected. CBTC is further extended to provide also k-connectivity with $\alpha \leq 2\pi/3k$ in [1]. A major advantage of CBTC and its extension is that the algorithms work in a distributed way.

An example in which we assume the nodes to have a disk graph transmission area is given in Figure 4.7. Note that the algorithm does not depend on the model used for transmission, assuming that the direction of signals is known. In Fig. 4.7(a) a node is assigned a transmission range that gives a maximal angle of $\beta_{max} = \pi$. So, if CTBC is applied to this node in the network the resulting local topology at u will be as depicted in Fig. 4.7(a) if α is bigger than π. An example for $\alpha = 1/2\pi$ is presented in Fig. 4.7(b). It can be seen easily in the figure that if the transmission power will be increased a bit, the maximal angle will still be $1/2\pi$. Therefore, a lot more power is needed to have a CBTC topology with a lower α.

XTC

The distributed algorithm XTC [53] is an algorithm that does not assume the wireless network to be a UDG, but a General Graph. To the best of the knowledge of the authors of [53], there is no other algorithm known to be working on General Graphs. It also does not assume to have location information, but works with link quality. Three steps can be distinguished: neighbor ordering, neighbor order exchange and edge selection. First, a node u sends with maximal power to detect its neighbors. They send a reply and this information is used by u to order the neighbors with respect to their link quality. Neighbors with a link with high quality are placed high in the list, the nodes with low quality



(a) A solution for the CBTC with $\alpha = \pi$.

(b) A solution for the CBTC with $\alpha = 1/2\pi$.

Figure 4.7: Two examples of CBTC with different α-values.

links low. Each node composes such a list and next all list are sent to their neighbors. So, a node u sends his list to all his neighbors, but receives also from each neighbor a list. The last step does not require any further communication. In this step node u traverses downwards through his ordered list (beginning with the neighbors with goods links) and selects the direct link to a node v if there is no "better" neighbor w that can be reached more easily from v than from u itself. In other words, the idea behind the algorithm is that u selects the paths from u to v among the direct path to v and all two-hop paths to v that maximizes the minimal link quality. So, for example if the link uv has quality 1 and uw and vw have quality 1.5 the path to v via w is preferred above the direct link uv, assuming that a higher quality value denotes a better quality.

Formally, we first create two empty sets: N_u and $\tilde{N}_u$. Then node u traverses through its list and we call v the least unprocessed neighbor in the ordered list of u. Node v will be added to $\tilde{N}_u$ if there exist a node w $in N_u \cup \tilde{N}_u$ (the processed nodes) such that the link quality of wv is better than uv. Otherwise it will be added to N_u. Next the following node in the ordered list of u is considered and finally $N_u \cup \tilde{N}_u$ is equal to the set of all neighbors of u. The neighboring edges of u in the final topology are those edges (u, v) of the graph G for which $v \in N_u$.

To give an impression of the created topology we present Figure 4.8.

4.3.2 Routing Algorithms

The base for an algorithm to set up a routing structure is a given topology. Therefore the topology influences the routing structure. Sometimes the topology even functions as a routing structure, see for example the Minimum Spanning Tree with the root located at a source node. In this section we discuss routing strategies that are less obviously connected to the topology. We focus again on broadcasting a message from one source to all nodes in the network.

Flooding

The strategy of flooding is to instruct every node to retransmit the messages it receives. But to avoid endless loops, only the first received instance of a message is retransmitted and all copies received later on are ignored.

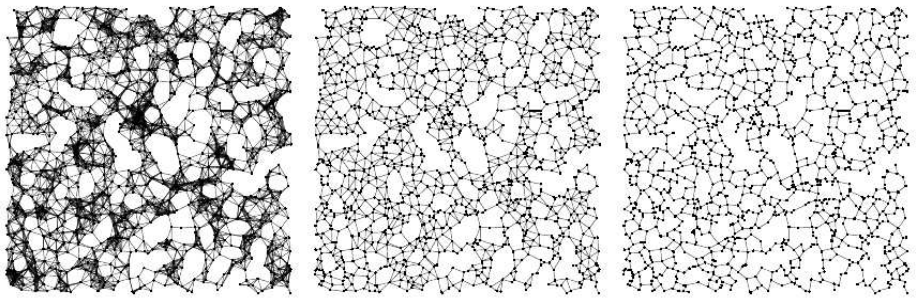


Figure 4.8: The Unit Disk Graph (left), the Gabriel Graph (center) and the graph after running XTC (right). The network contains 1400 nodes placed randomly and uniformly on a square field of 20 units side length. This Figure is taken over from [53].

In [37] the capability of flooding in high-mobile wireless networks is studied. An interesting outcome is that even flooding is insufficient for reliable broadcasting in such networks due to *flooding waves*, which we explain below. A message that is broadcasted spreads to neighboring nodes and neighbors of neighbors and so on. This spreading of a message can be seen as a flooding wave. If a node that not yet received the message moves to a position where it becomes a neighbor of nodes that already spread the message, it will not receive the message anymore. When nodes are moving fast, such situations do occur.

Another drawback of flooding is that it results in a storm of messages which can lead to interference. This is called the broadcast storm problem and is described in [47]. A solution to this problem is scoped flooding, i.e. a flooding protocol where some nodes relay the message and some do not, depending on a chosen factor like battery capacity, position or a randomly determined yes or no decision.

MPR Flooding

Multipoint Relay Flooding is a typical example of a scoped flooding algorithm. Only the Multipoint Relays have the task to retransmit the first received message and ignore the copies received later on. All the other nodes never retransmit a message. The basic idea behind MPR Flooding is that every node has its own MPRs among its neighbors and that these MPRs can be selected by looking at specific properties of the node. The nowadays used MPR selection algorithms are based on the MaxMin and MinTotal approach. MPR flooding is an improvement to flooding as it reduces the number of duplicate retransmissions. In Part II, Chapter 7 we discuss MPR flooding intensively, including the improvement compared to normal flooding.

BIP

One of the first algorithms on broadcasting in wireless network with usage of the wireless multicast advantage is the Broadcast Incremental Power algorithm (BIP) ([55]). We therefore spend slightly more words on BIP than on the other algorithms. BIP is an incremental algorithm that adds each step only one node.

	u_1	u_2	u_3	u_4	u_5
u_1	0	1	3.25	4.5	3.25
u_2	1	0	1.25	8.5	7.25
u_3	3.25	1.25	0	15.25	9
u_4	4.5	8.5	$15,25$	0	6.25
u_5	3.25	7.25	9	6.25	0

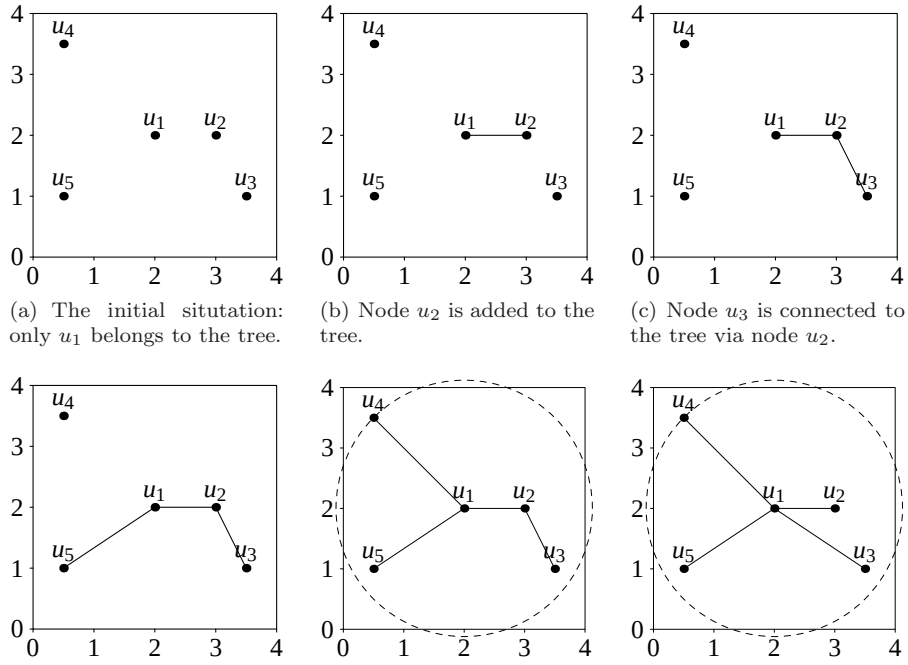
Table 4.3: The transmission power needed to send a message between two nodes in our example graph.

It begins with a tree T with only the source node. A node is to be added to the tree T if it is a neighbor of T and the additional power needed to include the node in the tree is minimum. We give an example situation in which we assume the power needed for transmission is quadratically related to the distance. In Figure 4.9(a) the source node (denoted by u_1) and the destination nodes (all the other nodes) are shown. The energy needed to transmit between the nodes is presented in Table 4.3.2. At the beginning the total costs (the sum of energy needed for transmission) of the tree is zero. The first node that will be added is node u_2 (Fig. 4.9(b)). This can be seen easily as it gives the minimum value in the first row of the table. Now we have two nodes in our tree T: u_1 and u_2 and the total transmit power is 1. For each of these nodes we compute the minimum additional energy to expand the tree. The nodes u_3 and u_5 can be reached by node u_1 with transmission energy 3.25, so the additional energy needed is $3.25 - 1 = 2.25$. This is larger than the additional cost for u_2 to reach u_3, which is 1.25. This leads to the conclusion that we have to select node u_3 via u_2 (Fig. 4.9(c)). In the next step u_5 is added via u_1 as this requires minimum energy to expand the tree (Fig. 4.9(d)). Finally, node u_4 becomes a member of the tree via u_5 (Fig. 4.9(e)). To optimize the tree further a *sweep* is performed, i.e. we remove the links that are redundant due to the wireless multicast advantage. In Figure 4.9(e) we see that node u_3 is reached directly by u_1. Therefore we can remove the link between u_2 and u_3 in order to get an optimized tree as shown in Fig. 4.9(f).

Variations on BIP Many research has been done to improve BIP by changing or adding elements. We discuss them shortly, as the basic elements do not change.

- *BIP(β):* BIP(β) is the short name of the Energy-Resource Limited Broadcast Incremental Power algorithm ([54]). It is based on BIP but has a modified cost function to discourage nodes with low battery residual capacity to join the broadcast tree. The cost function for each link uv is defined as follows: $C_{uv} = P_{uv} \left(\frac{E_u(0)}{E_u(t)}\right)^{\beta}$ in which C_{uv} is the cost for a link uv, P_{uv} the power needed to establish the link uv, $E_u(0)$ the initial energy value of node u, $E_u(t)$ the residual energy of u at time t and β a variable. For $\beta = 0$ BIP(β) is equal to the BIP algorithm.

- *BIPPN:* The Broadcast Incremental Power Per Node algorithm (BIPPN) is a node based MST heuristic as in [56] augmented with a sweep pass.



(a) The initial situtation: only u_1 belongs to the tree.

(b) Node u_2 is added to the tree.

(c) Node u_3 is connected to the tree via node u_2.

(d) Node u_5 is selected to join the tree, as it has the minimum incremental power needed to expand the tree.

(e) The last node to join the tree is u_4. The transmission range corresponding to the power needed to reach u_4 is shown by the dashed circle.

(f) After applying the sweep, the final routing tree is a star centered at u_1.

Figure 4.9: Example of constructing a broadcast tree using BIP.

Neighbors	Power Needed	Power per Neighbor needed
u_2	1	1
u_2, u_3, u_5	3.25	1.083
u_2, u_3, u_5, u_4	4.5	1.125

Table 4.4: Table with needed powers to reach one or more neighbors, as used in BIPPN.

The main difference is that BIPPN looks at the incremental power needed to reach more than one node instead of only one node. Look again at our example used for BIP. Starting with node u_1 BIPPN computes the energy needed per node to reach 1, 3, or 4 nodes, which is shown in Table 4.3.2. Note that since the distances between u_1 and u_3 and u_1 and u_5 are equal, there is no row for reaching just 2 neighbors. It can be seen easily that the minimum power per node is needed if we select u_2, u_3 and u_5 as neighbors. So in one step multiple nodes are added. The next step the same procedure is performed for all nodes in the tree (u_1, u_2, u_3, u_5) and the minimum value per node is selected again. When the algorithm has created a broadcast tree, a sweep can be done to possibly optimize the tree further.

- *BIPWLA:* In [40] the Broadcast Incremental Power With Look Ahead (BIPWLA) is presented. The algorithm can be seen as an extension for BIPPN and a tree is expanded in the following way. With a certain power assignment a number of nodes can be reached. Instead of looking at the power per neighbor needed, we can look at the number of neighbors for a specific power assignment. Let $u \in V$ be a node with an assigned power of $P(u)$. Then, the node to be added to the tree has to be the node u that maximizes $\frac{|N(u)|}{P(u)} + max\frac{|N(v)|}{Pv}|v \in N(u)$. So, we look not only at how expensive the selected node is, but also at the costs to add neighbors of the selected node. The algorithm is described in more detail in [40].

- *WBIP:* This variation of BIP is introduced in [22] and is an abbreviation of Weighted Broadcast Incremental Power. Essentially nothing is changed except for the edge weights. WBIP uses a so called Inverse Link Longevity as edge weight, which is the cost metric defined as the power needed for a transmission from u to v divided by the energy left in node u. This aspect aims more at maximizing the network lifetime than BIP.

MEN The Maximum Energy Node (MEN) heuristic ([40]) attempts to select nodes with sufficient battery residual capacity as nonleaf nodes in the broadcast tree. The algorithm expands a tree T as follows. In each round it begins with an empty Q. Every node u that is a leave of T, neighbored to a node v not in T and has enough energy to reach node v is added to the set Q. From this set Q the leaf node is selected that has maximum energy left and the nodes that can be reached with its energy are added to T. This procedure is followed till all nodes are within the tree. Of course a sweep pass can be applied to optimize the tree.

Part II

Multipoint Relay Flooding

Chapter 5

Introduction

The previous part showed some insight in the existing work and problems in the field of network lifetime in wireless networks. This part is completely addressed to one particular broadcast algorithm with the objective to maximize the network lifetime: Multipoint Relay (MPR) flooding. The broadcast strategy using MPRs has become very popular as it is intergrated in an open source protocol suitable for MANETs called Optimized Link State Routing protocol (OLSR). This popular protocol is standardized by the Internet Engineering TaskForce (IETF) and therefore the details of the protocol can be found in the corresponding Request for Comment [11].

The task of OLSR is to deliver data to all nodes such that routes between nodes can be computed with a routing algorithm like for example Shortest Paths. Although OLSR does not forward the messages itself, it is classified as a proactive routing algorithm, as it distributes the routing information. If a messages needs to be sent this can be done directly, without sending route requests to discover a route, as OLSR already gathered the necessary data to compute the best route. To have the right information available for computation of the routes, there is a continuous stream of control messages. These control messages contain information about a node, like position, energy residual et cetera, and have to be broadcasted to all nodes. This broadcasting is done by MPR flooding, which we discuss intensively in this part.

5.1 Research Problem Characterization

The classification methodology described in the first part of the thesis, can be used to specify the problem we study in this part. We describe shortly the assumed network properties, variables, objectives and the mechanism of the algorithm.

Network Properties In our situation we assume a stationary and layered wireless network, which has a flat structure. The nodes in th network a limited battery capacity. In this part of the thesis we assume that the nodes are placed randomly, with a uniform distribution, in a squared field or on a grid and that the nodes have a transmission area corresponding with a UDG or a General Graph. The assumed placement and the transmission area depend on the section. However, in each section we assume

bidirectional links. Failures in the network can only occur as a result of energy depletion, unexpected failures do not occur. We assume that nodes only use energy for transmitting a packet and not for receiving one. There is no location information or distance information used by the nodes to compute the routing structure. There are no assumptions about the detection model and time synchronization.

Variables Since we have a stationary network, the positions are fixed. The transmission powers are also fixed, but the traffic handling or routing structure may change during time. The type of communication is broadcasting with maximal one broadcast being processed by the nodes at each moment in time. So, new broadcasts are only sent if the previous broadcast has been finished.

Objectives MPR flooding is a simple algorithm that has the objective to reduce interference. However, our objective is to research methods to extend MPR flooding in such a way that the network lifetime is maximized (under the assumptions above). The network lifetime is defined as the time until the first nodes runs out of energy. MPR flooding is a local algorithm that uses only information of nodes in the 2-hop neighborhood and is therefore scalable. No special attention is given to the robustness of the algorithm. Since we assume the power assignment to be fixed and the routing to be variable in a layered network, the problem can be characterized as the SPDR problem, described in Section 3.2.3. There are no topological objectives, as the power assignment is fixed and MPR algorithms do not assign powers to nodes. There are also no objectives concerning the simplicity, data delivery ratio, sensing coverage, quality of surveillance, stealthiness or balanced energy consumption.

Algorithm Mechanism The selection of MPRs is a local procedure and therefore the algorithm is distributed.

5.2 Contributions

During the last few years several studies on MPR flooding have been published. However, to the best of our knowledge, this part of the thesis is the first more abstract study on MPR flooding. We present a theoretical study on MPRs, a study on MPR flooding and a study on MPR selection algorithms, instead of one global view on MPR flooding. The main advantage of our approach is that one can now clearly state the effect of the different parts in MPR flooding.

Using this approach we study each part separately and present theorems concerning the structure of the MPRs and the network lifetime for special networks. One of the aspects we show, is that in several situations the network lifetime cannot be improved and is fixed, independent of the used MPR selection algorithms, because of the definition of MPRs. This is the main theorem of the thesis.

With this outcome as a basis, we study the effect of MPR selection algorithm in several wireless network with different structures and show that the effect of an algorithm depends on a special property of the network, the so called 'Maximum Forcedness Ratio' of a network.

Another contribution is a new algorithm, which seems to be comparable with the best existing algorithm, even though it has not been optimized a lot. Furthermore, we present a MATLAB-simulator for testing and analyzing MPR selection algorithms and their effects on network lifetime if MPR flooding is used.

5.3 Structure

As mentioned in the previous section, we discuss each part of MPR flooding separately. First, in Chapter 6 the definition of MPRs and MPR sets are given. An MPR-set of a node is a subset of the neighbored nodes, which has as main characteristic, that if all nodes in the MPR-set of a certain node u relay the message, all 2-hop neighbors of u receive the message. This definition leads to several lemmas and theorems, especially if we assume the nodes to be located at a grid. Chapter 7 shortly discusses the correctness of MPR flooding and shows that a local algorithm can be used for global purposes. Chapter 8 studies the problem to select MPRs. Besides formal problem statements, existing selection algorithms with different goals, our new developed algorithm and also a theorem about the effect of the selection algorithms in networks where nodes lie on a grid are presented. The knowledge of MPRs is used in Chapter 9 to generate random networks and to simulate MPR flooding and analyze the results. The main goal of this chapter to relate the theorems to the simulations. In the final chapter, the work on MPRs is concluded and recommendations for further research are given.

Chapter 6

Multipoint Relays

In this chapter we present a formal mathematical analysis of Multipoint Relays (MPRs), based only on the definition of MPR-sets and an MPR. We deduce theorems for MPRs in two situations. In the first section we look at MPRs in general graphs and in the second section at MPRs in graphs with nodes located on a grid.

6.1 Definitions and Implications

6.1.1 Multipoint Relays and Multipoint Relay sets

The essential idea behind MPRs is that a guarantee is given that if the MPRs of a node relay the message of a source node, the complete 2-hop neighborhood receives the message sent by the source node. Therefore, MPRs define a local structure and not a global structure. In a communication protocol, MPR can be used in the way that if a certain node has to relay a message, all the nodes in its MPR-set also have to relay the message. We give the formal definitions of MPR-sets and MPRs in the following.

Definition An MPR-set of a node u is a subset $M(u) \subset N^1(u)$ which dominates $N^2(u)$. Furthermore, a node $v \in M(u)$ is called an MPR of a node u and the set of all possible MPR-sets of u is denoted by $MPR(u)$.

We only state that v is an MPR and omit the name of the selector node, if it is not of any interest to the situation. Of course, there has to be a node that selects v as MPR.

6.1.2 Variety in MPR-sets

In our analysis we study the degree in which MPR-sets can differ. For this goal we use the terms 'forced' and 'fixed' to define the variety in MPR-sets. The terms can be applied to nodes (forced and fixed nodes), but also to sets (forced and fixed sets). In this section we define these terms properly, first for nodes, then for sets.

Forced and Fixed Nodes

Definition The set $F^1(u) = \bigcap_{M \in MPR(u)} M$ is called the set of nodes forced by u to be MPR.

The meaning of the set $F^1(u)$ is the following. If a node u is part of the communication process, i.e. it has to relay a message, all nodes from $F^1(u)$ are also part of the communication, independent of the MPR-set chosen for u. In other words, they are forced to relay a packet sent by u.

We also use a definition that can be seen as the inverse of $F^1(u)$, as it defines the nodes that force u to be MPR.

Definition For a given node $u \in V$, the set of nodes that force u to be MPR, $F^{-1}(u)$, is defined as the set of all nodes v for which u has to be MPR in each possible MPR-set of v. In terms of the previous definition: $F^{-1}(u) = \{v | u \in F^1(v)\}$.

Both definitions can be related clearly by the following: $v \in F^1(u) \Leftrightarrow u \in F^{-1}(v)$. The following lemma gives a useful property for nodes u and v with $v \in F^{-1}(u)$.

Lemma 6.1.1 *$v \in F^{-1}(u)$ if and only if there exists a node $v^* \in N(u)$ such that there is only one path with 2 hops from v to v^*, being the path $v - u - v^*$. Furthermore, node v^* belongs also to the set $F^{-1}(u)$.*

Proof This lemma follows directly from the fact, that an MPR-set of a node has to guarantee that the 2-hop neighborhood of the node is dominated completely.

Suppose $v \in F^{-1}(u)$. Then there is a node $v^* \in N^2(v)$ which can be reached only via u. Otherwise, $N^1(v) \backslash u$ is also an MPR-set of v and dominates $N^2(v)$, which contradicts that u is forced to be an MPR. Consequently, there is only one 2-hop path between v and v^*. Since we have bidirectional links, $v^* \in N^2(v)$ means that also $v \in N^2(v^*)$. If we combine this with the fact that u is the only node that connects to both v and v^*, u has to be MPR for v^* in each possible MPR selection. Thus, $v^* \in F^{-1}(u)$.

The other way around, suppose that there exists a node $v^* \in N(u)$ such that there is only one path from v to v^* with 2 hops, being the path $v - u - v^*$ and suppose that $v^* \in F^{-1}(u)$. Then we can use the same proof if we only replace v by v^* and vice versa to prove that $v \in F^{-1}(u)$. □

We can interpret $F^{-1}(u)$ as follows. If one of the nodes in $F^{-1}(u)$ is part of a communication process, also node u is part of the communication process. So, u does not have a choice whether to relay a message or not, since it is forced by a node from $F^{-1}(u)$ to be an MPR. A special case occurs, when all neighbors of a node u force u to be MPR.

Definition A node u is called a *fixed MPR* if $F^{-1}(u) = N(u)$, i.e. if each neighbor of u forces u to be MPR.

An example for a fixed MPR is e.g. the center node in a star topology. In Figure 6.1 we present an example to clarify forced and fixed nodes further. A corresponding table in which $F^1(u)$ and $F^{-1}(u)$ are listed is presented in Table 6.1. From the table one can see that there is only one fixed node, namely node u_4.

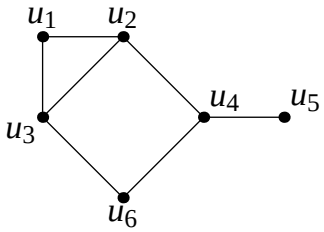


Figure 6.1: An example network with forced and fixed nodes.

Node u	$F^1(u)$	$F^{-1}(u)$
u_1	$\{u_2, u_3\}$	$\emptyset$
u_2	$\{u_4\}$	$\{u_1, u_4\}$
u_3	$\emptyset$	$\{u_6\}$
u_4	$\{u_2\}$	$\{u_2, u_5, u_6\}$
u_5	$\{u_4\}$	$\emptyset$
u_6	$\{u_3, u_4\}$	$\emptyset$

Table 6.1: A list of the relation to force a node to be MPR, for the example network in Figure 6.1.

Forced and Fixed Sets

The definitions of $F^1(u)$ and $F^{-1}(u)$ for a node u can be extended to make the definitions suitable for sets U. Using this definitions we also can define a fixed MPR-set instead of a fixed MPR-node.

Definition $U = u_1, u_2, \ldots, u_n$ and let $MPR(U) = \{M(u_1) \cup M(u_2) \cdots \cup M(u_m) | M(u_i) \in MPR(u_i)\}$ be the set of all possible combinations of MPR-sets of nodes $u \in U$. Then $F^1(U) = \bigcap_{M \in MPR(U)} M$ is called the set of nodes forced by U to be MPR.

Definition For a given $v \in V$, a set $U \subset N(v)$ is forced for node v, if for each possible MPR-set $M(v)$, we have $M(v) \cap U \neq \emptyset$. By $F^{-1}(U)$ we denote the set of all nodes which force the set U.

Note that we do not restrict the set U in the definition of $F^1(U)$, but do restrict the set U to be a subset of a neighborhood of a node v in the definition of $F^{-1}(U)$. Furthermore note, that the definition of F^{-1} says that set U is forced by a set V if all nodes in V use at least one node of U as MPRs instead of that all nodes in V use all nodes in U as MPR. Because of this, the definitions of $F^1(U)$ and $F^{-1}(U)$ are not the inverse of each other anymore.

With the above given definitions, we can define fixed MPR-sets as follows.

Definition A set U is called a *fixed MPR-set* if $F^{-1}(U) = (\bigcup_{u \in U} N(u)) \backslash U$, i.e. if for all nodes v which are neighbored to node of U, the set U is forced by v.

To explain the forced and fixed sets further, we use the example network that is shown in Figure 6.2. Suppose we have a set $U = \{u_2, u_6\}$, then $F^1(U) = \{u_4\}$

and $F^{-1}(U) = \{u_3, u_4\}$, and therefore is U a fixed set. To show that the function is not invertible anymore, we take $U = \{u_3, u_4\}$ and see that $F^1(U) = \emptyset$ instead of the set $\{u_2, u_6\}$, which would be the case if the function is invertible. Note however, that $F^{-1}(U) = \{u_2, u_5, u_6\}$, so U is again a fixed set. Not all sets are fixed in this network. Look for example at the set $U = \{u_5, u_6\}$, for which $F^1(U) = \{u_4\}$ and $F^{-1}(U) = \emptyset$.

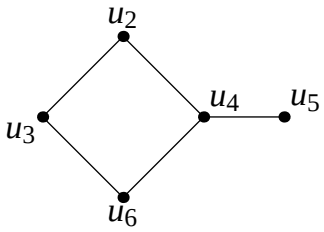


Figure 6.2: An example network with forced and fixed sets.

6.2 Graphs on Grids

In the previous sections we did not assume special network structures. However, if we restrict the network to be of a special type, more information can be used to deduce properties which can lead to interesting theorems about MPRs. A special type of graph that is considered in this section is the type in which nodes are located on grid points and a disk graph model is used for the transmission range of a node. We first give definition, then list properties of the network and finally present some lemmas and theorems on MPRs on grid graphs.

6.2.1 Properties of Grids

A grid (see Figure 6.3 for a 3 by 4 grid) is characterized by its regular structure, from which two important properties can be deduced. These two properties are the basic elements in the proofs given in this chapter.

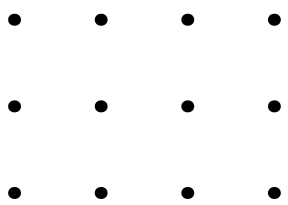


Figure 6.3: A 3 by 4 grid.

Translation Property If u is a grid point and $u + \overrightarrow{a}$, with $\overrightarrow{a}$ a vector, is also a grid point, then $v + \overrightarrow{a}$ is a grid point if v is a grid point. We call this the translation property.

Symmetry Property If v is a grid point, then the nodes obtained by mirroring v in the vertical, horizontal and both diagonal lines through another

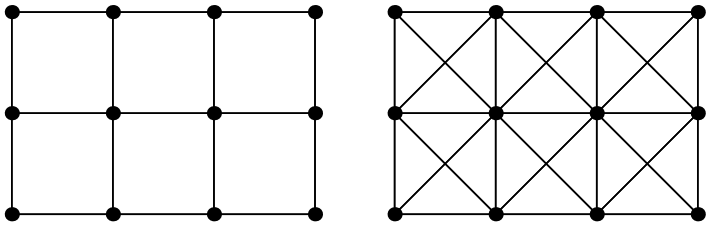
grid point are also grid points. Combining the horizontal and vertical symmetry, we get point symmetry in u, and therefore this property is called the symmetry property. The following definition gives a useful notation that can be used to denote the mirror image of a point.

Definition Given points u and v (not necessarily grid points) and a vector $\overrightarrow{a} := v - u$, we can define the point v mirrored in u, denoted by $\overline{v}^u$, as $\overline{v}^u = u - \overrightarrow{a}$.

6.2.2 Definitions for Grids on Graphs

Definition We denote by $G_{m\times n}(r)$ a graph with $m \cdot n$ nodes on grid points of a $m \times n$ grid. We assume the horizontal and vertical distance between neighboring grid points to be 1. Two nodes u and v are connected by an edge in $G_{m\times n}(r)$ if and only if $d_g(u, v) \leq r$ (Disk Graph property).

For example, $G_{m\times n}(1)$ (see Figure 6.4(a)) has only horizontal and vertical edges and $G_{m\times n}(\sqrt{2})$ (see Figure 6.4(b)) has also edges between diagonal neighboring grid points. Note, that $G_{m\times n}(1)$ is what normally is considered as a grid graph.



(a) The graph $G_{3\times 4}(1)$. (b) The graph $G_{3\times 4}(\sqrt{2})$.

Figure 6.4: Two graphs on grids.

Since we are not interested in the side-issues at the border of the graph, we have to look at nodes in the center of the graph. The 1-hop neighborhood of such a central node is point symmetrical in the central node. This is obviously not true for a border node. In the next definition we clarify what we mean with a central node and a border node.

Definition Nodes in the graph $G_{m\times n}(r)$ that are at located in the $(m-2\lfloor r\rfloor)\times(n-2\lfloor r\rfloor)$ *subgrid* that is created by removing $\lfloor r\rfloor$ grid rows from the upper and lower side of the grid and $\lfloor r\rfloor$ grid columns from the left and right side of the grid are called *1-hop central nodes*. Nodes in $G_{m\times n}(r)$ that are not central nodes, so the removed nodes, are called *border nodes*.

In some of our proofs we need the 2-hop neighborhood of a node to be point symmetrical. Therefore we introduce a second definition to describe the nodes that satisfy our needs.

Definition Nodes in the graph $G_{m\times n}(r)$ that are at located in the $(m-4\lfloor r\rfloor)\times(n-4\lfloor r\rfloor)$ *subgrid* that is created by removing $2\lfloor r\rfloor$ grid rows from the upper and lower side of the grid and $2\lfloor r\rfloor$ grid columns from the left and right side of the grid are called *2-hop central nodes*.

Obviously, the set of 2-hop central nodes is a subset of the set of 1-hop central nodes.

6.2.3 Properties of Graphs on Grids

One of the properties of graphs $G_{m\times n}(r)$ is that multiple values for r can produce the same edge set. Formally, every transmission range r is part of a domain $[r_b, r_e)$ such that for every $r^* \in [r_b, r_e)$ the graph $G_{m\times n}(r^*)$ is identical to the graph $G_{m\times n}(r)$ (for an example, see Figure 6.5). This property is basically due to the discrete nature of grid points in contrary to the radius that can be chosen from $\mathbb{R}$.

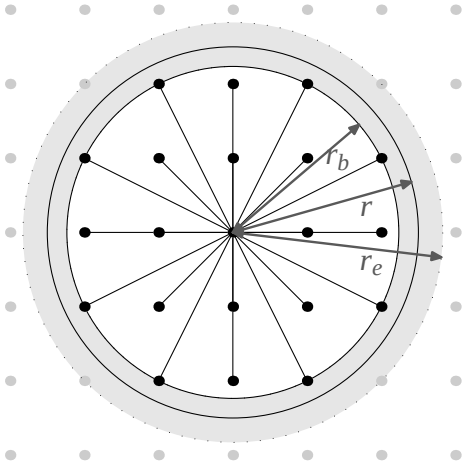


Figure 6.5: The light gray colored area depicts the domain $[r_b, r_e)$, such that for every r in this domain $G_{m\times n}(r)$ is identical to $G_{m\times n}(r_b)$.

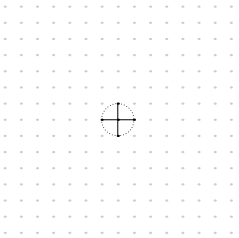
A second property can be deduced from the symmetry property of a grid. If u has a neighbor v, then the Euclidean distance between u and v is less than r. The Euclidean distance between u and $\overline{v}^u$ is also less than r and consequently $\overline{v}^u$ is also a neighbor of u. Similarly, the node given by mirroring v only on the horizontal, vertical or diagonal line through u is also a neighbor of u. The neighbor set of u can therefore be called highly symmetrical.

To give an idea of the structure of the graphs $G_{m\times n}(r)$, we depict in Figure 6.6 the local neighborhood of 1-hop central nodes of the first twelve grid graphs when increasing the transmission range r.

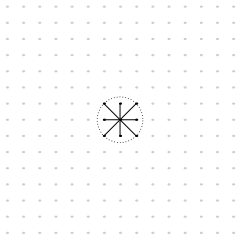
6.2.4 MPRs in Graphs on Grids

In this section we present our mathematical analysis of MPRs in graphs $G_{m\times n}(r)$. We use the properties defined for grids and graphs on grids and the earlier definitions, to present formal theorems and proofs.

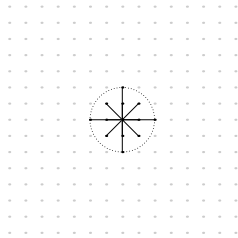
Theorem 6.2.1 *For every 1-hop central node $u \in G_{m\times n}(r)$ with $r \geq 1$, there are at least 4 nodes in $N(u)$ that force u to be an MPR. If one of the nodes for which u has to be MPR is not located on the symmetry axes, i.e. the horizontal,*



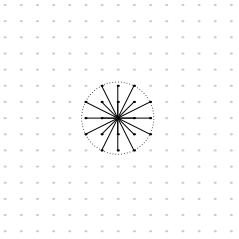
(a) The 1-hop neighborhood of a node with transmission range equal to $\sqrt{1}$.



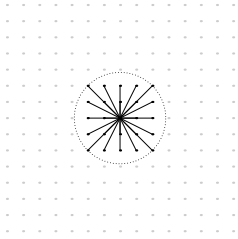
(b) The 1-hop neighborhood of a node with transmission range equal to $\sqrt{2}$.



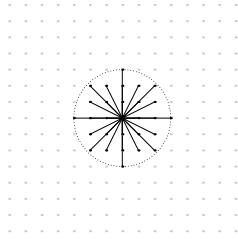
(c) The 1-hop neighborhood of a node with transmission range equal to $\sqrt{4}$.



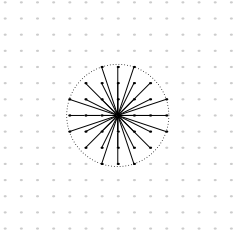
(d) The 1-hop neighborhood of a node with transmission range equal to $\sqrt{5}$.



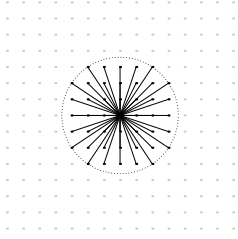
(e) The 1-hop neighborhood of a node with transmission range equal to $\sqrt{8}$.



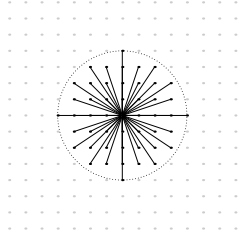
(f) The 1-hop neighborhood of a node with transmission range equal to $\sqrt{9}$.



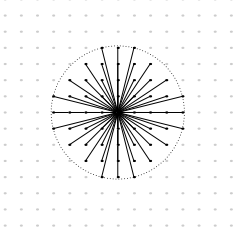
(g) The 1-hop neighborhood of a node with transmission range equal to $\sqrt{10}$.



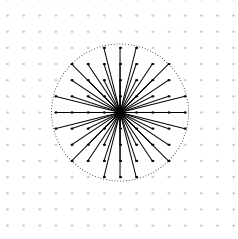
(h) The 1-hop neighborhood of a node with transmission range equal to $\sqrt{13}$.



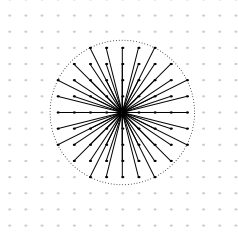
(i) The 1-hop neighborhood of a node with transmission range equal to $\sqrt{16}$.



(j) The 1-hop neighborhood of a node with transmission range equal to $\sqrt{17}$.



(k) The 1-hop neighborhood of a node with transmission range equal to $\sqrt{18}$.



(l) The 1-hop neighborhood of a node with transmission range equal to $\sqrt{20}$.

Figure 6.6: Symmetry of 1-hop neighborhoods of graphs on grids with several transmission ranges.

vertical and both diagonal axes, then there are at least 8 *nodes in* $N(u)$ *that force* u *to be an MPR.*

Proof Let $[r_b, r_e)$ be the equivalence domain of radii producing the graph $G_{m\times n}(r)$. By the equivalence relation we can take the smallest radius, r_b, which means that there is at least one grid point $v \in N(u)$ that lies on the circle with radius r_b. Due to the symmetry property in graphs on grids, there are at least 4 nodes located on the circle with radius r_b if v is on the horizontal, vertical or diagonal line through u. If v is not located at one of these mentioned lines, then there are least 8 nodes located on the circle with radius r_b. Each node v_i located on the circle, has a 2-hop neighbor on the other side of the circle, denoted by $\overline{v_i}^u$. We now claim that u is forced to be MPR for v_i.

If there is no other 2-hop path from v_i to $\overline{v_i}^u$, node u is forced to act as an MPR by Lemma 6.1.1. In Figure 6.7 the 2-hop path $v_i - u - \overline{v_i}^u$ is given. As the distance between v_i and $\overline{v_i}^u$ is equal to the diameter of the circle with radius r_b, there is no other node that can reach both v_i and $\overline{v_i}^u$ (see Fig. 6.7), which proves the claim and, thus, also the theorem. □

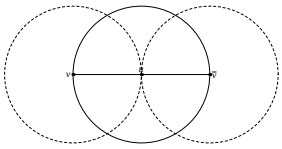


Figure 6.7: The only node that can be reached by v and its mirror image $\overline{v}^u$ is u. So, also only one path from v to $\overline{v}^u$ exists.

Remark Not all nodes in $F^{-1}(u)$ have to lie on the circle with radius r_b. Take for example a node from the graph $G_{m\times n}(3)$. As shown in Figure 6.8, there are 4 nodes of $F^{-1}(u)$ on the circle and 4 are inside the circle. The nodes in $F^{-1}(u)$ can be described as all nodes v and $\overline{v}^u$ in which u is the only grid point in the intersection of the disk graphs of u and v.

If a node v has only one 2-hop path to $\overline{v}^u$ it clearly belongs to $F^{-1}(u)$ (see Lemma 6.1.1). In the next lemma, we prove that also the reverse is true.

Lemma 6.2.2 *Let* u *be a 2-hop central node in* $G_{m\times n}(r)$ *with* $r \geq 1$*, then* $v \in F^{-1}(u)$ *if and only if* $\overline{v}^u$ *is the only node in* $N(u)$ *with only one 2-hop path to* v*.*

Proof If $\overline{v}^u$ is the only node in $N(u)$ with only one 2-hop path to v, then $v \in F^{-1}(u)$ by definition of $F^{-1}(u)$.

Now, suppose $v \in F^{-1}(u)$ and assume that there is a node $v^* \in N(u)$ with $v^* \neq \overline{v}^u$ such that there is only one 2-hop path between v and v^*, namely via u. By the symmetry of the graph we know that both $\overline{v}^u$ and $\overline{v^*}^u$ exist and are connected to u. The difference between v^* and $\overline{v}^u$ can be described by the vector $\overrightarrow{a} := \overrightarrow{v^* - \overline{v}^u}$. We get $\overline{v^*}^u + \overrightarrow{a} = (u - (v^* - u)) + (v^* - \overline{v}^u) = u - (\overline{v}^u - u) = v$. Let us now define $c := u + \overrightarrow{a}$. Since $|\overrightarrow{a}|$ is certainly less than $2r$ and u is a 2-hop central node, c also exists in the graph and $c \in N^2(u)$. Thus, the path $v - c - v^*$ exists (see Figure 6.9), which contradicts our assumption. Therefore

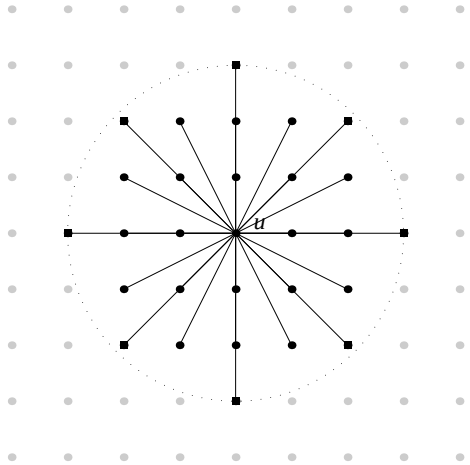


Figure 6.8: The nodes that force u to be a MPR, i.e. $F^{-1}(u)$, are depicted by little squares. From $F^{-1}(u)$ there are 4 nodes on the circle with radius r_b and 4 in the circle.

$v^* = \overline{v}^u$. In other words, $\overline{v}^u$ is the only node in $N(u)$ with only one 2-hop path to v. □

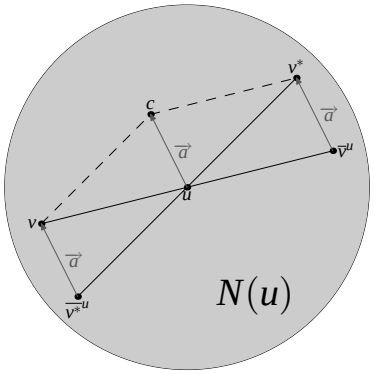


Figure 6.9: Sketch for the proof of Lemma 6.2.2.

The symmetry property of the grid can be used to give a sort of symmetrical relation on forcing a node. If a node v forces a node u, node u also forces node v. We state this properly in the next lemma.

Lemma 6.2.3 *Let u be a 2-hop central node in $G_{m\times n}(r)$ with $r \geq 1$, then $v \in F^{-1}(u)$ if and only if $u \in F^{-1}(v)$.*

Proof Suppose $v \in F^{-1}(u)$, then by Lemma 6.2.2 there is only one 2-hop path between v and $\overline{v}^u$, being via u. By the translation property of the grid, there is also only one 2-hop path between u and $\overline{u}^v$, being via v. We know that this node $\overline{u}^v$ exists in the graph $G_{m\times n}(r)$, since u is a 2-hop central node. Using again Lemma 6.2.2, $u \in F^{-1}(v)$.

To prove it the other way around, we can switch u and v and use the same method of reasoning. □

We intend to give a geometrical characterization of $F^{-1}(u)$. For this goal we introduce the convex hull and the extreme points of the convex hull around $N(u)$. We show in Corollary 6.2.6 that a node is in $F{-}1(u)$ if and only if it is an extreme point of u. For this corollary we need two lemmas, which are presented below.

Definition A convex hull around $N(u)$, denoted by $C(u)$, is the smallest convex space in which all nodes in $N(u)$ are located.

Definition Let C be a convex subset of a vector space X. A point $x \in C$ is called an extreme point if it is not an interior point of any line segment in C. That is, x is extreme if and only if whenever $x = ty + (1-t)z$, $t \in (0,1)$, $z \neq y$, implies either $y \notin C$ or $z \notin C$.

Lemma 6.2.4 *Let u be a 2-hop central node in $G_{m\times n}(r)$ with $r \geq 1$, then all nodes in $F^{-1}(u)$ are extreme points of the convex hull $C(u)$.*

Proof We proof the lemma by contradiction. Let $f \in F^{-1}(u)$ and h be the line orthogonal to $u - f$ through point f. Assume g is a point on h or in the half space defined by h which does not contain u and that $g \in N(u)$. Furthermore, let $\overrightarrow{a} = \overrightarrow{(f-g)}$, yielding $f - \overrightarrow{a} = f - \overrightarrow{(f-g)} = g$. The point obtained by mirroring g in f is given by $\overline{g}^f = f + \overrightarrow{a}$ and also belongs to $N(u)$. Since f, g and $\overline{g}^f$ are in $N(u)$, these nodes mirrored in u are also in $N(u)$ and we denote them respectively by $\overline{f}^u$, $\overline{g}^u$ and $\overline{\overline{g}^f}^u$. See Figure 6.10(a) for a sketch.

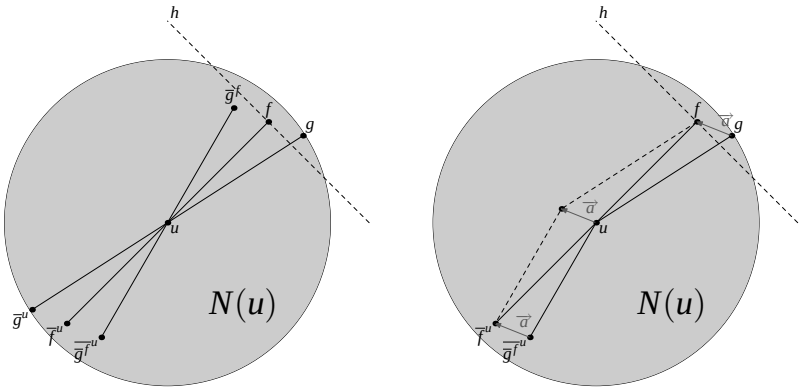
Due to symmetry we can use $\overline{g}^f = f + \overrightarrow{a}$ to deduce that $\overline{\overline{g}^f}^u = \overline{f}^u - \overrightarrow{a}$. As there exists an edge between u and g, i.e. $d_g(u,g) < r$, there is also an edge between $u + \overrightarrow{a}$ and $g + \overrightarrow{a} = f$. The same holds for u and $\overline{\overline{g}^f}^u$, such that there is also an edge between $u + \overrightarrow{a}$ and $\overline{\overline{g}^f}^u + \overrightarrow{a} = \overline{f}^u$ (see Figure 6.10(b)).

So, if there exist a node $g \in N(u)$ on h or in the half space defined by h which does not contain u, then there exists a path from f via $(u + \overrightarrow{a})$ to $\overline{f}^u$ and a path from f via u to $\overline{f}^u$. Applying Lemma 6.2.2, node f can not be in $F^{-1}(u)$. This is a contradiction and therefore our assumption has to be wrong. Therefore, there is no node g on h or in the half space defined by h which does not contain u. Consequently, f is an extreme point of $C(u)$. □

Lemma 6.2.5 *Let u be a 2-hop central node in $G_{m\times n}(r)$ with $r \geq 1$, then every node $v \in N(u)$ with $v \notin F^{-1}(u)$ is an interior point of a line segment in $C(u)$.*

Proof Let v be a node with $v \in N(u)$ and $v \notin F^{-1}(u)$. Consequently, there are at least two 2-hop paths to $\overline{v}^u$ (cf. Lemma 6.2.2) and therefore at least one 2-hop path besides the path $v - u - \overline{v}^u$. Assume that this path is the path from v via a node $w = u + \overrightarrow{a}$ to $\overline{v}^u$. By translation, there is also a path from v via $\overline{w}^u = u - \overrightarrow{a}$ to $\overline{v}^u$.

Let z be the node such $z = v + \overrightarrow{a}$ and $\overline{z}^v = v - \overrightarrow{a}$ (see Figure 6.11). Since $d_g(u,z) = d_g(\overline{v}^u, w) \leq r$ and $d_g(u,\overline{z}^v) = d_g(w,v) \leq r$, we know that $z, \overline{z}^v \in N(u)$. As v is an interior point of the line segment between z and $\overline{z}^v$ and both nodes are in $N(u)$, v is an interior point of a line segment in $N(u)$. Consequently, v is also an interior point of $C(u)$, the convex hull of $N(u)$. □



(a) A sketch of the situation described in the proof of Lemma 6.2.4.

(b) The existence of a path besides $f - u - \overline{f}^u$.

Figure 6.10: Explanation for the proof of Lemma 6.2.4.

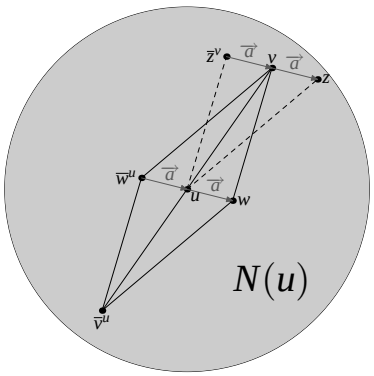


Figure 6.11: Explanation for the proof of Lemma 6.2.5.

Corollary 6.2.6 *Let u be a 2-hop central node in $G_{m\times n}(r)$ with $r \geq 1$, then node $v \in N(u)$ is in $F^{-1}(u)$ if and only if v is an extreme point of the convex hull $C(u)$.*

Proof Suppose $v \in N(u)$ is in $F^{-1}(u)$. By Lemma 6.2.4 v has to be an extreme point of the convex hull $C(u)$.

Now, suppose v is an extreme point of the convex hull $C(u)$. Lemma 6.2.5 proves that a node is an interior point of a line segment in $C(u)$ if the node is not in $F^{-1}(u)$. Therefore v has to be in $F^{-1}(u)$ to be an extreme point of the convex hull $C(u)$. □

By definition, $C(u)$ consists of all nodes in $N(u)$. In the next lemma we prove that the double of convex hull $C(u)$ consists of all nodes in the $N^2(u)$. In the following we define what we mean with the double of a convex hull $C(u)$.

Definition Let $C(u)$ be the convex hull around $N(u)$, with the extreme points $F^{-1}(u) = (f_1, f_2, \ldots, f_n)$. The double of the convex hull $C(u)$, denoted by $C_2(u)$ is then defined by the convex hull with extreme points $F_2^{-1}(u) := (u + 2\overrightarrow{(f_1 - u)}, u + 2\overrightarrow{(f_2 - u)}, \ldots, u + 2\overrightarrow{(f_n - u)})$.

Lemma 6.2.7 *Let u be a 2-hop central node in $G_{m\times n}(r)$ with $r \geq 1$, then for every node $w \in N^2(u)$ it holds that $w \in C_2(u)$.*

Proof Assume without loss of generality that u is located at the origin. If $w \in N^2(u)$, then there is a node $v \in N(u)$ such that $w \in N(v)$. By the definition of a convex hull around a neighborhood of a node, $v \in C(u)$ and $w \in C(v)$. Let $(f_1, f_2, \ldots, f_n)$ be the extreme points of $C(u)$. Note that due to translation $(w + f_1, w + f_2, \ldots, w + f_n)$ are the extreme points of a node w, if w is a 1-hop central node. Since every point in a convex hull can be described as a convex combination of the extreme points of the hull, v can be written as $v = \sum_{j=1}^{n} \lambda_j f_j$ with $\sum_{j=1}^{n} \lambda_j = 1$ and $\lambda_j \geq 0$.

Using translation, node $w \in C(v)$ can be written similarly as $w = \sum_{k=1}^{n} \mu_k (f_k + v) = \sum_{k=1}^{n} \mu_k (f_k + \sum_{j=1}^{n} \lambda_j f_j)$ with $\sum_{j=1}^{n} \lambda_j = 1$, $\sum_{k=1} \mu_k = 1$ and $\lambda_j, \mu_k \geq 0$. Rewriting the equation, leads to the following:

$$\begin{aligned} w &= \sum_{k=1}^{n} \mu_k (f_k + \sum_{j=1}^{n} \lambda_j f_j) \\ &= \sum_{k=1}^{n} \mu_k f_k + \sum_{k=1}^{n} \left(\mu_k \sum_{j=1}^{n} \lambda_j f_j \right) \\ &= \sum_{k=1}^{n} \mu_k f_k + \sum_{j=1}^{n} \lambda_j f_j \\ &= \sum_{j=1}^{n} (\mu_j + \lambda_j) f_j \\ &= \sum_{j=1}^{n} \frac{\mu_j + \lambda_j}{2} 2 f_j \end{aligned} \tag{6.1}$$

Since $\sum_{j=1}^{n} \frac{\mu_j+\lambda_j}{2} = 1$ and $\frac{\mu_j+\lambda_j}{2} \geq 0$, equation 6.1 states that w can be written as a convex combination of $(2f_1, 2f_2, \ldots, 2f_n)$, so w is in $C_2(u)$. □

The next theorem states that every node in $N^2(u)$ is neighbor of a node in $F^{-1}(u)$. So, $F^{-1}(u)$ is an MPR-set of u.

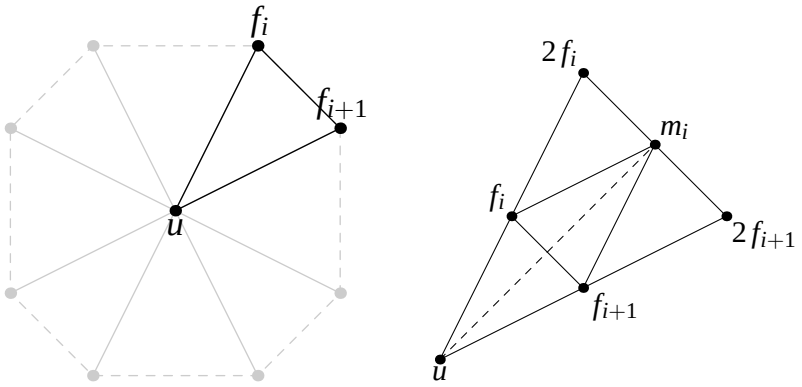
Theorem 6.2.8 *For every 2-hop central node $u \in G_{m\times n}(r)$ with $r \geq 1$, the set $F^{-1}(u)$ is an MPR-set of u, i.e. $F^{-1}(u) \subset MPR(u)$.*

Proof By Lemma 6.2.7 every node $w \in N^2(u)$ is in $C_2(u)$. So, if we prove that all grid points in $C_2(u)$ are the grid points defined by $\bigcup_{f\in F^{-1}(u)} N(f)$, we can deduce that $N^2(u) = \bigcup_{f\in F^{-1}(u)} N(f)$. In the proof we use $\triangle a, b, c$ to denote a triangle with corner nodes a, b and c.

Let $f_i, f_{i+1} \in F^{-1}(u)$ be clockwise neighboring extreme points of $C(u)$, as depicted in Figure 6.12(a). We prove that the grid points in $\triangle u, 2f_i, 2f_{i+1}$, which is a subset of $C_2(u)$, are in $N(f_i)$ or $N(f_{i+1})$ (see Figure 6.12(b)). We use m_i to denote the middle of $2f_i$ and $2f_{i+1}$.

In general, if a node u reaches p_1 and p_2, u can reach all grid points in $\triangle u, p_1, p_2$, since the triangle lies in the circle of radius $max(d_g(u, p_1), d_g(u, p_2))$ centered at u. Since node f_i is the middle of u and $2f_i$ and $u \in N(f_i)$, also $2f_i \in N(f_i)$. As node f_{i+1} is the middle of u and $2f_{i+1}$ and u reaches f_{i+1}, we know by translation that f_i reaches m_i. Thus, node f_i reaches nodes $2f_i, m_i$ and u and therefore we can use the statement to deduce that it reaches all grid points in $\triangle f_i, 2f_i, m_i$ and $\triangle f_i, m_i, u$. Using the same argumentation, f_{i+1} reaches all grid points in $\triangle f_{i+1}, 2f_{i+1}, m_i$ and $\triangle f_{i+1}, m_i, u$. So, every grid point in $\triangle u, 2f_i, 2f_{i+1}$ is in $N(f_i)$ or $N(f_{i+1})$.

Therefore, all grid points in $C_2(u)$ are in the union of $N(f)$ with $f \in F^{-1}(u)$. Since all grid points in $N^2(u)$ lie in $C_2(u)$, the statement $N^2(u) = \bigcup_{f\in F^{-1}(u)} N(f)$ holds. □



(a) The triangle formed by u, f_i and f_{i+1}, which is a subset of the convex hull $C(u)$.

(b) The triangle formed by $u, 2f_i$ and $2f_{i+1}$, which is a subset of the convex hull $C_2(u)$.

Figure 6.12: Triangles in $C(u)$ and $C_2(u)$.

Chapter 7

MPR Flooding

The local communication structure defined by Multipoint Relays can be used in a global way, by flooding a message via the MPRs. This is called MPR flooding and is used in the OLSR protocol defined in the introduction in Chapter 5. The difference between normal flooding and MPR flooding is discussed in the first section. Section 7.2 proves the correctness of MPR flooding. The last section uses the results of the previous chapter to compute upperbounds on the network lifetime for stationary wireless networks.

7.1 Flooding versus MPR flooding

Although we described flooding and MPR flooding in the first part in Chapter 4, we summarize the main issues. Normal flooding is the communication strategy in which every node relays the first instance of a message. All copies received later on, are ignored. In MPR flooding a node u only relays an incoming message if it is the first time the message is received and the last hop of the message is a node v for which $u \in M(v)$, i.e. u is in the MPR-set of v. The pseudo-code of an MPR flooding algorithm is given in Algorithm 1. The principal difference between normal flooding and MPR flooding is that in normal flooding all neighbors of a node have to relay a message and in MPR flooding only designated neighbors, that are in the MPR-set of the node, have to relay the message. This reduces the number of (duplicate) transmissions and reduces consequently the interference problems. Figure 7.1 gives an example of the reduction. For the MPR Flooding in Figure 7.1(b), the center node selects the black colored nodes as MPRs. These nodes relay the packet, as they receive the packet for the first time. The MPR selection of a black node consists of the center node and two of the uncolored neighbors of the center node. However, these nodes do not retransmit the packet, as the center node is the source of the packet and the the other nodes receive the packet for the second time.

7.2 The Correctness of MPR Flooding

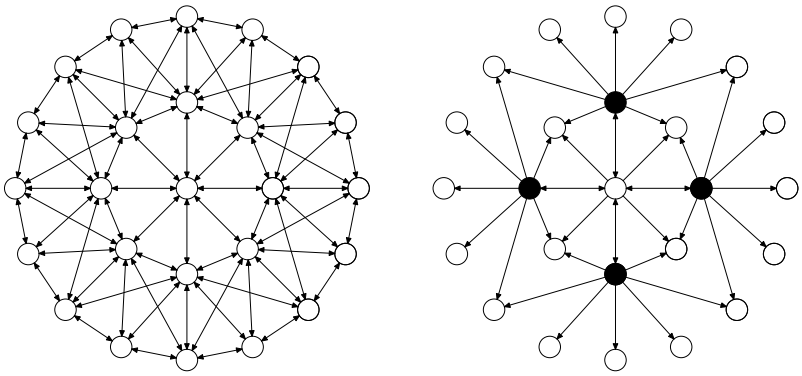
In this section we assume a finite and connected graph with error free transmissions between two neighboring hops, enough battery capacity to be available for every node to retransmit messages and a General Graph as transmission model

Algorithm 1 The procedure for MPR flooding

Require: a connected graph $G(V, E)$, for each node $v \in V$ one selected MPR-set $M(v) \in MPR(v)$ and a node v_0 that initiates a broadcast message m in the network.

Ensure: All nodes in the network receive the message m

```
repeat
  for each node u ∈ V who receives the message for the first time do
    if u ∈ M(v) with v the node from which u received the message then
      u sends message m to all of its neighbors, i.e. to N(u)
    end if
  end for
until no nodes exists, which receive the message for the first time
```



(a) Flooding a packet in a wireless multihop network. All transmissions are shown.

(b) Flooding a packet in a wireless multihop network from the center node using MPRs. Only the black colored nodes relay the packet. All transmissions are shown.

Figure 7.1: Pure flooding versus MPR flooding.

and prove the correctness of MPR flooding, i.e. we prove that all nodes are reached by MPR flooding. If every MPR relays the message, Theorem 7.2.1 proves the fact. However, it may happen that some nodes get the message from a node, for which they are MPR, but do not send the message, since they already got the same message earlier, however, from a node for which they were not an MPR. This case needs some further investigation and is treated in Lemma 7.2.2. First, we look at the case in which every MPR relays the message.

Theorem 7.2.1 *If all MPRs relay a message, a broadcast message initiated at v_0 arrives at all destinations.*

Proof Since we assume error free transmission, we know that if v_0 sends a message $N(v_0)$ receives the message. Node v_0 has selected a MPR-set $M(v_0)$ such that the set of neighbors of $M(v_0)$ covers the $N^2(v_0)$. Furthermore, nodes in $N^2(v_0)$ are selected as MPR such that the set of neighbors of MPRs covers the $N^3(v_0)$. Generalizing, in each $N^k(v_0)$ there are selected MPRs such that the neighbor set of the MPRs covers the $N^{k+1}(v_0)$. Consequently, since we assume all MPRs to relay the message it is guaranteed that if $N^k(v_0)$ receives a message, also $N^{k+1}(v_0)$ receives the message. Starting with $N^1(v_0)$ we see by induction that all nodes receive the message, as every node in a connected network is element of a set $N^p(v_0)$ for some $p \in \mathbb{N}$. □

The situation looks different if not all MPRs relay the message, since we can not state anymore directly that the 1-hop neighborhood of an MPR-set is reached. We therefore first describe how such a situation can occur and then prove that the 1-hop neighborhood of an MPR is still reached if the MPR does not relay the message.

As mentioned earlier, MPRs only relay the first instance of a message and ignore copies. For this purpose every node maintains a duplicate set, in which all received messages are listed. This set is used to check if an incoming message already has been processed. If so, the message will be ignored. Consequently, if a node receives a message from a node for which it is MPR it will only retransmit the message if the message is not listed in the duplicate set. By this behavior it is possible that a node receives a message from a node for which it is not an MPR and therefore will not retransmit it and later on receives the same message from a node for which it is MPR. This message will also not be retransmitted as the message already has been processed.

In the next lemma we prove formally that all the nodes in the neighborhood of an MPR still receive the message even though the MPR did not retransmit the message.

Lemma 7.2.2 *Every 1-hop neighborhood of an MPR is still reached if duplicate messages are ignored.*

Proof Let u be an MPR for some node v, i.e. $u \in M(v)$. We prove that $N^1(u)$ is reached by looking at two cases: u is a relaying MPR (including the case that u is a source node) and u is a non-relaying MPR. The case for which u is not an MPR is not interesting as u is not responsible for the deliverance of a message to its neighborhood.

Case 1: u is a relaying MPR and by the assumption of error free transmissions we know that $N^1(u)$ receives the message if u transmits a message.

Case 2: Assume that this situation is the first in time in which an MPR does not relay a message. Node u must have at least one neighbor for which it is an MPR (v_1) and one for which it is not an MPR (v_2) and that has sent the message earlier than v_1 (see Figure 7.2). This results in the situation that u is still an MPR for v_1, but does not relay the message.

The fact that node u is an MPR of v_1 means that u is needed to reach some nodes in $N^2(v_1)$. As u can only reach its neighbors, u is selected as MPR for v_1 to reach a subset of $N(u)$ or the complete $N(u)$.

Since u is not an MPR of v_2, v_2 has selected other MPRs to reach $N^2(v_2)$. We know for sure that these MPRs relay the message, as u is the first non-relaying MPR and v_2 sent the message before v_1. Consequently it is guaranteed that the complete $N(u)$, which is a subset of $N^1(v_2) \bigcup N^2(v_2)$, is reached. Therefore, the next moment in time can be seen again as the first moment in which it is doubtful whether the neighborhood of a non-relaying MPR receives the message. Applying the same reasoning, we can prove by this inductive method that the neighborhood of all non-relaying MPRs is reached.

Conclusion: In both cases it is proven that $N(u)$ is reached. □

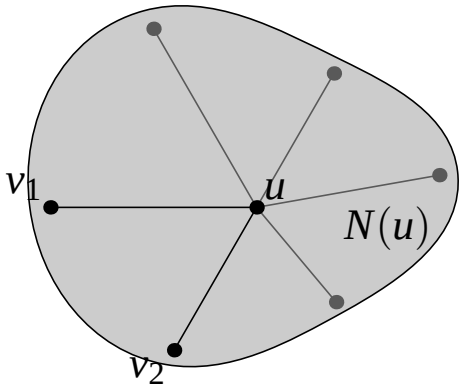


Figure 7.2: An example to clarify handling duplicate messages.

Using this lemma, we can adjust the proof of Theorem 7.2.1 slightly to prove that MPR flooding reaches all destination, also if MPRs relay only messages that are received for the first time. Since the essence of the proof does not change, we give a short version of the proof.

Theorem 7.2.3 *If MPRs only relays the first instance of a message and ignore copies, a broadcast message initiated at v_0 arrives at all destinations.*

Proof Node v_0 reaches all its neighbors directly. Each $N^k(v_0)$ selects MPRs that have neighbors that cover $N^{k+1}(v_0)$. Using Lemma 7.2.2 we can state that the the complete neighborhood of an MPR-set receives the message, even if some MPRs do not relay the message. Since the network is connected, the message reaches all nodes in the network. □

7.3 Network Lifetime

In the following, we assume that broadcasts are initiated such that there is a fixed period between every two subsequent broadcasts. Therefore, the network lifetime can be expressed in the number of messages that have been correctly broadcasted. In this section we analyze the network lifetime if MPR flooding is used and deduce upperbounds on the network lifetime for network with fixed MPRs or fixed MPR-sets.

7.3.1 Network Lifetime in Networks with Fixed Nodes

The network lifetime of a network can be bounded from above if there exists a fixed node in the network. This upperbound is given in Corollary 7.3.2, but first we present a theorem on a fixed MPR.

Theorem 7.3.1 *Let u be a fixed MPR in a connected network. Then u is used exactly once for transmission for each broadcast message in the network if MPR flooding is used for communication.*

Proof We observe two cases: u being the source node and u being not the source node. If u is the source, it obviously transmits a message. If u is not the source, the message has to reach node u via its neighborhood. Since $F^{-1}(u) = N(u)$, u is MPR for every neighbor. The first message that arrives at u is therefore being relayed and the duplicate messages are ignored. So, in both cases u transmits a message exactly once which proves the theorem. □

Corollary 7.3.2 *Let G be a connected network graph with a set $S \neq \emptyset$ of fixed MPRs. Furthermore, let the initial battery capacity of a node u be denoted by $E_0(u)$ and the battery cost per transmission of a message be a constant $C_t(u)$ for a nodes u in G. Then, if MPR flooding is used for communication, the number of correctly finished broadcast messages, which is equal to the network lifetime, denoted by $NLT(G)$, is bounded by:*

$$NLT(G) \leq \left\lfloor \min_{u \in S} \frac{E_0(u)}{C_t(u)} \right\rfloor \tag{7.1}$$

Assuming that the transmission costs are the only cost, the equality holds, if all transmission costs are equal and the initial battery capacities of the non-fixed nodes are equal or bigger than $\min_{u \in S} E_0(u)$.

Proof Every broadcast message that is sent trough the network reduces the batteries of a node $u \in S$ exactly once with $C_t(u)$. If a node $u \in S$ does not have enough energy anymore to transmit a message, the number of successful broadcasts will not increase, as u is necessary as MPR to reach the complete network. The node u with the lowest quotient of the initial battery capacity and the energy consumption per transmission will be the first node that can not transmit a message anymore. This node u can send $\frac{E_0(u)}{C_t(u)}$ messages. As the number of completed broadcasts is an integer, the result has to be floored. If there is node in the network that has a lower battery value than node u or high transmission costs, this node may run out of energy earlier than u. Therefore, $\frac{E_0(u)}{C_t(u)}$ is an upperbound.

If, however, the transmission costs are equal and these are the only costs, only the initial battery capacity determines how many messages can be transmitted. When the non-fixed nodes have equal or more initial battery capacity than node u, node u will certainly be the first node that runs out of energy. □

7.3.2 Network Lifetime in Networks with Fixed Sets

For networks with fixed sets we can develop an expression for the network lifetime in a similar way as done for a network with fixed nodes. However, in the previous analysis we used the fact that each fixed node is used exactly once for transmission for each broadcast, which we cannot use in this situation, as we do not know which nodes in U are used as relays. We only know that at least one node in U is used as a relay.

Theorem 7.3.3 *Let U be a fixed MPR-set in a connected network. Then for every broadcast in the network at least one node in U is used for transmission if MPR flooding is used for communication.*

Proof We observe two cases: U containing the source node and u containing not the source node. If U contains the source, it obviously transmits a message. If U does not contain the source, the message has to reach set U via its neighborhood, i.e. $(\bigcup_{u\in U} N(u))\backslash U$. Since $F^{-1}(U) = (\bigcup_{u\in U} N(u))\backslash U$, there is a node in U that relays the first message that reaches u. So, in both cases at least one node in U transmits a message which proves the theorem. □

Corollary 7.3.4 *Let G be a connected network graph with a set $S \neq \emptyset$ of fixed MPR-sets. Further more, let the initial battery capacity of a node u be denoted by $E_0(u)$ and the battery cost per transmission of a message be a constant $C_t(u)$ for a node u in G. Then, if MPR flooding is used for communication, an upperbound on the number of correctly finished broadcast messages and, thus, on the network lifetime $NLT(G)$ is given by:*

$$NLT(G) \leq \min_{\overline{S}\in S} \left(\sum_{u\in\overline{S}} \left\lfloor \frac{E_0(u)}{C_t(u)} \right\rfloor \right) \tag{7.2}$$

Proof For a fixed set $\overline{S} \in S$, the nodes in $\overline{S}$ can in total transmit at most $\sum_{u\in\overline{S}} \left\lfloor \frac{E_0(u)}{C_t(u)} \right\rfloor$ times. The set $\overline{S}$ that minimizes the number of messages that can be transmitted is therefore the upperbound on the network lifetime. □

7.3.3 Computational Costs

Corollary 7.3.2 is a special case of Corollary 7.3.4, but has far less computational costs. To give the minimum upperbound on the number of messages that can be sent over a network using Corollary 7.3.2, one has to check for every node u if its neighborhood set is equal to the set that forces u, $F^{-1}(u)$. Next, a simple computation gives the upperbound. Corollary 7.3.4 demands that we do this for every possible subset U, in which the only restriction is that U has to be a subset of a neighborhood of some node. Obviously, checking every possible subset of U has far more computational costs.

Chapter 8

MPR Selection

Although we have shown in Chapter 6 that there are networks in which forced MPRs exist, mostly there are still situations in which one can choose a node from a neighbor set to join the MPR-set. To illustrate this, we present an example. Suppose we have a network as depicted in Figure 8.1. In this situation there are three MPR-sets for u_1, from which one has to be selected. The first MPR-set is the set that consists of node u_2, so $M_1(u) = \{u_2\}$, the second MPR-set is defined by $M_2(u) = \{u_4\}$ and the third MPR-set is the set $M_3(u) = \{u_2, u_4\}$. All these sets satisfy the definion for being an MPR-set of u_1.

In this chapter we describe several algorithms that select an MPR-set from all possible MPR-sets. We first formulate two possible selection problems and give complexity results. Then, selection algorithms are discussed, including the algorithm we propose. Finally a theorem is given that states that each MPR selection algorithm selects the same MPR-set for a 2-hop central node in a graph $G_{m\times n}(r)$.

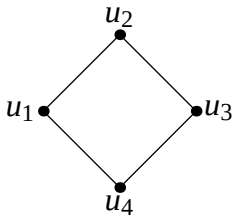


Figure 8.1: Selecting MPRs.

8.1 MPR Selection Problems

In this section we formally describe two MPR selection problems, namely selecting the MPR-set of minimum cardinality and selecting the MPR-set with minimum total costs. Other MPR selection problems, like selecting the MPR-set with minimum overlapping, are described in [36]. However, we are not aware of any literature for the problem of selecting the MPR-set with minimum maximal cost or maximum minimal costs.

8.1.1 Selecting the MPR-set of Minimum Cardinality

The classical MPR selection problem is to find a set of MPRs of minimum size that covers the whole 2-hop neighborhood. The formal problem definition of the *Minimum Multipoint Relay* problem is as follows.

Instance: A network G (defined as a graph $G(V, E)$), a node u of $V(G)$ and and integer B.
Question: Is there an MPR-set of u with size less than B?

Selecting the MPR-set of minimal cardinality has been proven to be NP-complete ([48], [42]). The idea behind the proofs is that the Minimum Set Cover problem can be reduced by a simple polynomial reduction to the problem of selecting the MPR-set of minimum cardinality. The Set Cover problem has been proved to be NP-complete by Karp ([24]).

8.1.2 Selecting the MPR-set with Minimum Total Costs

If all nodes are assigned equal powers, then the problem to select the MPR-set with minimum cardinality corresponds with the goal to select the MPR-set with minimum total power. However, in some networks the powers are not equally distributed, or other costs have to be minimized. For these situations it is useful to look at the problem to select the MPR-set with minimum total costs.

Instance: A network G (defined as a graph $G(V, E)$), a node u of $V(G)$, costs c_v for all $v \in V(G)$ and and integer B.
Question: Is there a MPR-set of u such that the total cost of the MPR-set is less than B?

Like for the Minimum MPR problem, there is also a Set Cover problem that can be reduced to the problem to select the MPR-set with minimum total costs: the Weighted Set Cover problem. This problem is the NP-complete problem to select subsets from a set such that the subsets cover the complete set and have minimum total weight. Selecting the MPR-set with minimum total cost is therefore also NP-complete. However, we are not aware of any formal proofs of the NP-completeness of the problem to select the MPR-set with minimum total costs.

8.2 Existing Heuristic Algorithms for Selecting MPRs

Due to the complexity results in the previous section, heuristic algorithms are used to select MPR-sets. Just like the other algorithms discussed in Chapter 4, these MPR selection algorithms also use a MinTotal or MaxMin approach. We describe in this section some of the existing algorithms.

Minimum Cardinality MPR Selection

The implementation of MPR selection as described in [42] uses a rather simple incremental algorithm to compute an MPR-set. To the best of our knowledge, this is the first MPR selection algorithm in literature. In the following, we describe the selection procedure for a node u.

1. The first nodes that are selected as MPRs are the nodes in $N(u)$ that are the only neighbor of a node in $N^2(u)$.
2. While there are still uncovered nodes in $N^2(u)$: select the nodes from $N(u)$ as MPRs that are neighbor to the largest set of uncovered nodes from $N^2(u)$. In case of ties the node is taken with the largest value of $d_u^+(v) = |\{w \in N(v) | v \in N(u) \text{ and } w \in N^2(u)\}|$, i.e. the number of neighbors of a neighbor v of u that are two hops away from u.
3. When the condition of the while-loop in step 2 is not met anymore, the final optimization can be applied. This optimization discards any MPR node v such that the MPR-set excluding v still covers the whole neighborhood $N^2(u)$.

An example of this construction is given in Figure 8.2. This algorithm chooses first nodes as MPRs with many neighbors in $N^2(u)$, with the intention to get an MPR-set with minimum cardinality. Therefore this algorithm can be seen as a algorithm using the MinTotal approach with respect to the number of MPRs.

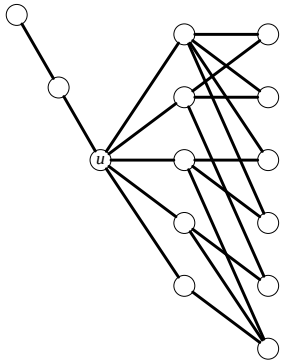
In the rest of the thesis we refer to this algorithm by using the abbreviation MinCar.

Maximum Willingness MPR Selection

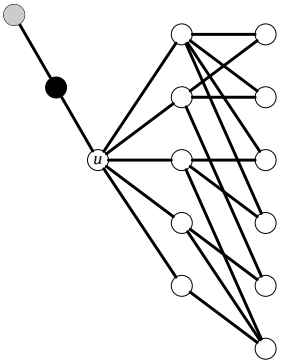
An extension to the above described algorithm is the heuristic used for the implementation of OLSR in RFC3626 [11]. Since not all nodes are the same, there can be a difference in willingness to be a relay node. Take for example the battery capacity. To address this problem, the willingness of a node is taken into account. In OLSR eight values are available for the willingness going from 0 ("will never") to 7 ("will always").

The algorithm to construct a set of MPRs taking into account willingness is the following.

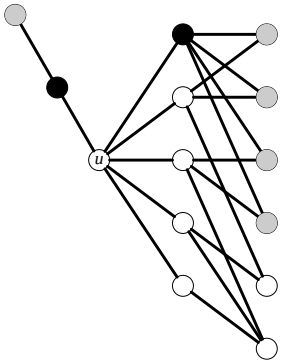
1. All nodes from $N(u)$ that are the only neighbor of a node in $N^2(u)$ or have a willingness equal to "will always"are selected as MPR.
2. While there are still uncovered nodes in $N^2(u)$: select the nodes from $N(u)$ with the highest willingness that cover at least one uncovered node of $N^2(u)$. In case of multiple choices, select the node with the most uncovered neighbors in $N^2(u)$. If ties exist, the node is taken with the largest value of $d_u^+(v) = |\{w \in N(v) | v \in N(u) \text{ and } w \in N^2(u)\}|$, i.e. the number of neighbors of a neighbor v of u that are two hops away from u.
3. As an optimization, process each MPR with willingness smaller than "will always"in increasing order of willingness and check if the $N^2(u)$ is still covered if the processing node is excluded from the MPR set. If so, the MPR node may be removed.



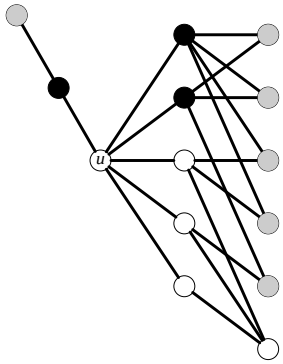
(a) The initial situation with node u as a local node that selects its MPRs.



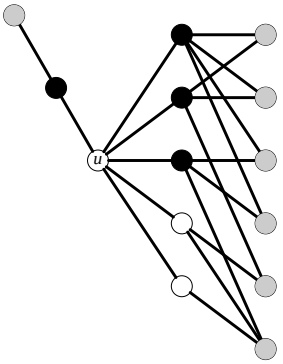
(b) There is exactly one 2-hop neighbor of u that has only one neighbor, which causes a selection of an MPR.



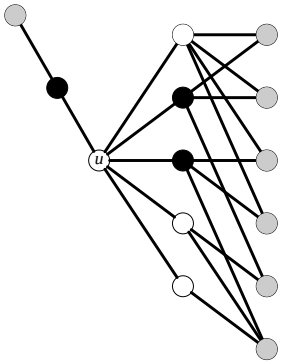
(c) The 2-hop neighborhood is not yet covered, so we look at the node that is neighbor to the largest set of uncovered neighbors.



(d) Still, the 2-hop neighborhood is not fully covered. However, there are now two nodes with the same number of uncovered neighbors. It can be easily seen that the node depicted higher, is connected to more 2-neighbors (covered and uncovered) than the one depicted lower.



(e) Only one 2-hop neighbor is not covered, therefore the last MPR is selected.



(f) Node u can discard an MPR without nodes becoming uncovered. This is also the final MPR selection produced by the heuristic.

Figure 8.2: Constructing a set of MPRs belonging to node u following the heuristic in [42]. Black nodes denote MPRs, gray colored nodes the nodes that are covered by the MPRs.

This MPR selection algorithm selects first the nodes with the highest willingness and saves by this selection criterion nodes with a low willingness. So, the MPR strategy seems to be in the category to maximize the minimum willingness of all nodes in the network. The consequence of this approach can be that more than the minimum number of required nodes are selected to be MPR.

We use the abbreviation MaxWill for this MPR selection algorithm.

8.2.1 E-OLSR:1 and E-OLSR:2

In [18] adjustments are made to the MPR algorithm as described in Section 8.2. The first extension (E-OLSR:1) changes the tie breaking rule in the MPR selection in Step 2. Instead of looking at $d_u^+(v)$ the node is picked with the most energy left. Available energy is the main discriminating rule in E-OLSR:2, that adjusts Step 2 of the original algorithm. Instead of adding nodes with maximum coverage, nodes from $N(u)$ are added in the order of available energy: nodes with full batteries are added before nodes with low batteries. If nodes have the same amount of residual battery, the original process can be run to break ties.

Although the authors present their algorithms as adjustments to OLSR, it is not fully clear what their improvements are. One possibility is that they define the willingness values by the battery values, but this is not really an improvement to OLSR, but more an example of how the willingness can be used.

8.2.2 Other Heuristics

More variation on the original heuristic for MPR selection can be found in [36]. Most of them only adjust the discriminating rules in Step 2, similar to E-OLSR:1 and E-OLSR:2.

8.3 New MPR Selection Algorithm

In simulations of MPR flooding, as discussed in the next chapter, networks in which the MPR-sets are selected by MaxWill MPR selection seem to have a better network lifetime compared to networks in which the MPR-sets are selected by the MinCar algorithm. Therefore, we take the basic elements of MaxWill and try to improve the algorithm. In this section we present our developed algorithm.

8.3.1 Maximum Willingness and Minimum Forced MPR Selection

The MaxWill MPR selection algorithm uses the willingness information of the nodes in the network to improve the network lifetime. In OLSR eight different states are used and therefore a simple improvement is to increase the number of states by defining the willingness of a node to be equal to the residual energy of the node. This of course may lead also to more computational effort for the nodes to select MPRs, but we now only look for improvements to the network lifetime and do no worry too much about computational effort. By the increased number of states, nodes can better decide which MPRs have to be

selected. Although this improves the results of the algorithm, it is not really an improvement to the algorithm as it does not change an essential element of the algorithm. The real change we propose is described in the following.

In the mathematical analysis in the previous sections we proved that fixed nodes have to relay the message once for each broadcast message. For fixed nodes we can not change anything about that, but if a node is not forced by all its neighbors, we probably can do something. Suppose we have a node v that is forced by many of its neighbors, than it presumably has to relay messages for many broadcasts. Therefore, it seems preferable to let the neighbors of v that do not force v to be MPR, select other nodes as MPR. So, what we intend to do is that a node u selects its MPRs from $N(u)$ based on how much a node $v \in N(u)$ is forced. We can define how much a node is forced by the following ratio: $\frac{|F^{-1}(v)|}{|N(v)|}$.

This additional element can be helpful for improving the network lifetime. It provides a look-ahead on the residual energy in the future, as it describes the expectation that a node will consume much energy as it is used by many nodes as MPR. We combine this approach with the willingness based on residual energy in the following Maximum Willingness and Minimum Forced MPR Selection algorithm, which we abbreviate by MaxWillMinForced.

1. All nodes from $N(u)$ that are the only neighbor of a node in $N^2(u)$ are selected as MPR.

2. While there are still uncovered nodes in $N^2(u)$: select node v from $N(u)$ that has the highest value $s(v) = E(v)\left(1 - \frac{|F^{-1}(v)|}{|N(v)|}\right)$ and covers at least one uncovered neighbors of $N^2(u)$. The term $E(v)$ denotes the residual energy of a node v. In case of multiple choices, select the node with the most uncovered nodes of $N^2(u)$. If ties exist, the node is taken with the largest value of $d_u^+(v) = |\{w \in N(v) | v \in N(u) \text{ and } w \in N^2(u)\}|$, i.e. the number of neighbors of a neighbor v of u that are two hops away from u.

3. As an optimization, process each MPR with willingness smaller than "will always"in increasing order of willingness and check if the $N^2(u)$ is still covered if the processing node is excluded from the MPR set. If so, the MPR node may be removed.

The value of $s(v)$ lies in the interval $[0, E(v)]$ and is the outcome of a mixed objective, based on the residual energy and how much node v is forced. If v is a fixed node, $s(v) = 0$ and if v is not forced to be MPR by any of its neighbors, $s(v) = E(v)$. In the next chapter we describe the performance of this algorithm compared to other algorithms.

8.4 The Effect of the Discriminating Step in MPR Selection Algorithms

The mathematical analysis in the previous chapters are the fundament for the main theorem about MPR selection algorithms for grid graphs. Before presenting the theorem we discuss the structure of MPR selection algorithms.

8.4.1 Algorithms

Although there are differences between the MPR selection algorithms, they all have a structure that can be divided into three parts. We therefore call this type of algorithms *three-step* algorithms.

1. Start with an empty MPR-set of node u, denoted by $M(u)$, and add nodes of $N(u)$ that are the only neighbor of a node in $N^2(u)$. So, after this step $M(u) = F^{-1}(u)$.
2. While there are still uncovered nodes in $N^2(u)$, select the nodes from $N(u)$ using some kind of decision making formula, which depends on the MPR selection algorithm.
3. Optimize the MPR-set by discarding any node in $M(u)$ and checking if the $N^2(u)$ is still dominated completely. The order of nodes to discard differs per MPR selection algorithm

For every MPR selection algorithm, the MPR-set contains at least the nodes are forced to be MPR. Consequently, Step 1 of the three-step algorithmic structure is an element of each MPR selection algorithm. A procedure to optimize the selected MPR-set, as in Step 3, can be expected to be also an element of an algorithm. However, there is some variation possible in the order of nodes which are considered for discarding. Between the initial step, Step 1, and the final step, Step 3, there is an intermediate step, Step 2, that selects non-forced MPRs by some criteria. These criteria vary per MPR selection algorithms and therefore this step characterizes the MPR selection algorithm.

8.4.2 MPR Selection Algorithms in Graphs on Grids

For the theorem in this section, we look again at the special network structure in which the nodes of a graph are located on a grid as defined by $G_{m\times n}(r)$. This theorem is our main theorem of the study on MPR selection in graphs $G_{m\times n}(r)$. It states that only forced nodes determine the MPR-sets for 2-hop central nodes in a graph.

Theorem 8.4.1 *For every 2-hop central node $u \in G_{m\times n}(r)$ with $r \geq 1$, u selects $F^{-1}(u)$ as its MPR-set if a three-step MPR selection algorithm is used.*

Proof Since the nodes in $F^{-1}(u)$ force u to be MPR, u forces each of the nodes in $F^{-1}(u)$ to be MPR by Lemma 6.2.3. Theorem 6.2.8 states that $F^{-1}(u)$ dominates $N^2(u)$ completely. As step 2 of the algorithm only is processed if there are uncovered nodes in $N^2(u)$, there are no nodes added to the MPR-set and therefore $F^{-1}(u)$ is selected as MPR-set of u. Step 3 can not remove any of the nodes in the MPR-set as they are all forced to be MPR. □

This theorem implies that one should not use 2-hop central nodes in graphs of type $G_{m\times n}(r)$ to test the effects of different discriminating rules in Step 2 of the MPR selection algorithm, since the MPR-sets in these graphs are fixed by using only Step 1. The observation also point out that the graph structure has a strong effect on the MPR selection algorithms. We discuss this further in the next chapter, in which also simulations are presented.

Chapter 9

Simulations

Even though the mathematical analysis in the previous chapters gives good insight in MPR flooding, it mainly provides theorems about 2-hop central nodes in specific grid networks. Since in practice the networks are not that regular and we are interested in the network lifetime of the whole network, including the border nodes, we use simulations to explore MPR flooding further.

In this chapter we generate random networks, simulate MPR flooding and measure certain properties, like the network lifetime, the number of fixed nodes and the so called 'Maximum Forcedness Ratio', which is defined later. The most important result of the chapter is that the Maximum Forcedness Ratio may be used to explain the effect a MPR selection algorithm has on the network lifetime.

The structure of this chapter is as follows. In the first section we describe the basics of our simulations and in the second section we report on simulations to investigate five topics. In Section 9.2.1 we focus on how many fixed nodes exist in random networks, since these nodes are crucial for the network lifetime. Afterwards, in Section 9.2.2, we provide some simulations to show that the number of fixed nodes increases linearly if the number of nodes in the network and the area of the squared field in which the nodes are placed increase linearly. Next, we investigate in Section 9.2.3 the 'Maximum Forcedness Ratio', which describes 'how fixed' the 'most fixed' nodes in a network are. The last two topics are about the differences in network lifetime by using different MPR selection algorithms in MPR flooding. In Section 9.2.4 we study the performance differences in graphs $G_{m\times n}(r)$ for several values of m,n and r and relate the results to Theorem 8.4.1. Finally, in Section 9.2.5 we discuss the intuitive idea that the 'more fixed' the 'most fixed' node in a network is, the smaller the performance difference is between two MPR selection algorithms.

9.1 Simulation Description

In Part I we presented a classification methodology and stated that characterizing network properties is necessary to compare algorithms fairly. This statement hold also for simulations and their properties. Therefore, we describe in this section some aspects of the simulation and the simulation settings.

9.1.1 Simulator

For the simulations a simulator is needed that can be used for simulating MPR selection and MPR flooding in wireless networks. Among all available simulators, the most interesting and often used simulators in MANETs are GloMoSim/QualNet, ns-2, OMNeT++, OPNET. In Appendix B short descriptions of these simulators are given. Although it seems preferable to choose one of the existing simulator, we decided to build our own simulator. The main reasons are the complexity of the existing simulators and the simplicity of MPR selection and MPR flooding. Building our own simulator makes it easy to satisfy all our wishes without having extra (disturbing) options. Details on the built event based simulator (MPRSimulator) can be found in Appendix A.

9.1.2 Simulation

The simulations are based on the problem description in the introduction. So, we simulate stationary networks with bidirectional links and use a UDG for the transmission model. This is the framework, in which we simulate as follows.

We create a random network by placing nodes randomly in a squared field using a uniform distribution. In this network, all nodes are assigned the same initial power and energy. Then, the nodes select MPRs using a predefined MPR selection algorithm. During the simulation, the nodes select new MPRs if node properties in their 2-hop neighborhood change. Such a property can be the residual energy of a node, the neighborhood of a node or the position of a node. The selected MPR-set of a node u is communicated to $N(u)$ such that nodes that belong to $M(u)$ are instructed to relay messages of node u. We assume that there are no costs involved for this communication.

In the network, messages have to be broadcasted. We use a sendpattern of broadcasts, in order to have exactly the same simulation for two different algorithms. By this we can compare algorithms fairly. The sendpatterns are created such that all nodes initiate a broadcast one after another and that all previous broadcasts have been completely processed before a new broadcast is initiated. Therefore, the network lifetime in a simulation can be expressed as the number of broadcasts.

In grids graphs $G_{m\times n}(r)$, the order of the nodes that initiate a broadcast, according to the sendpattern, can be described by the following, in which we use the notation (i, j) to denote the node at position (i, j). The first broadcast is initiated by $(1, 1)$. After the broadcast by (i, j), $(i, j + 1)$ initiates a broadcast, unless $j = n$. If $j = n$, the next node to broadcast in the sendpattern is $(i+1, j)$. However, if $i = m$, then the first node, $(1, 1)$, initiates again a broadcast and the order of the nodes is repeated. For randomly generated networks, there is also an order of nodes that is repeated in the sendpattern. However, this order is chosen randomly and not in a systematical way like the order in the grid.

We assume only the transmitting and not the reception of a message to reduce the energy of a node. So, in the complete simulation a node only consumes energy by transmitting a message and nothing else.

The simulation stops if one node in the network has run out of energy. The collected information during the simulation is then used to compute the resulting characteristics like the network lifetime et cetera.

In each simulation, all nodes have equal initial battery capacities, set at

100, and equal transmission ranges. We assume the transmission costs to be independent of the transmission range and equal to 5.

9.1.3 Result Variables

For analyzing the performance differences between several algorithms, we need measures that quantify aspects of the given instances or the outcome of the simulation. We describe the used resulting characteristic briefly.

Number of Fixed Nodes In Section 7.3.4 we presented a formula that gives the network lifetime if there is a fixed node in the network. We are therefore interested in the number of fixed nodes to see if such situations occur often in random graphs.

Maximum Forcedness Ratio When fixed nodes determine the network lifetime, the performance of MPR flooding is determined, independent of the MPR selection algorithms. We therefore like to see if 'almost' fixed nodes also influence the difference in performances of several MPR selection algorithms that are used for MPR flooding. As a fixed node is forced to be MPR by all its neighbors, we introduce the *Forcedness Ratio* to define 'how fixed' a node is. For a node u the Forcedness Ratio is defined as $\frac{|F^{-1}(u)|}{|N^1(u)|}$. Obviously, the Forcedness Ratio equals 1 if a node is fixed.

Since each node has a Forcedness Ratio, we can take the maximum of all Forcedness Ratios to determine how fixed the most fixed node in a network is. Evidently, the Maximum Forcedness Ratio is also equal to 1 if there is a fixed node in the network.

Network Lifetime Since the broadcast messages are sent regularly we can count the number of completely processed broadcasts to compute the Network Lifetime.

9.2 Simulation Results

This section presents the results of the characteristics and simulations that may help to get more insight in the effect of the network structure on algorithms. We focus on the number of fixed nodes in random networks, the scalability of fixed nodes, the Maximum Forcedness Ratios in random networks, the performance differences of MPR selection algorithms in graphs on grids and the effect of the Maximum Forcedness Ratio of a network on the performance of the MPR selection algorithm.

9.2.1 Number of Fixed Nodes

Generation

The number of fixed nodes is a property of the graph and is not related to the MPR selection algorithm. In Corollary 7.3.2 we proved that the network lifetime is determined by a fixed node if the transmission costs and the initial battery capacity are equal for each node. Therefore, we want to know how often such situations occur. For the generation of the networks, we have chosen to use the

same parameters as in [36]. For each network one has to choose the number of nodes from the set $[50, 100, 150, 200, 250, 350, 400]$ and the transmission range for all nodes from the set $[200, 250, 300, 350]$. The nodes are then placed in a field of 1000 by 1000 units. By this combination of different number of nodes and transmission ranges we get 28 different network settings.

For each network setting we create 100 random networks and analyze the structure of the network by counting the number of fixed nodes and the Maximum Forcedness Ratio. For analysis we look at the mean, the standard deviation, the maximum and the minimum of the result variables of the 100 created networks.

Results

The results are shown in Tables 9.1-9.4. The tables respectively present the mean number, standard deviation, minimum number and maximum number of fixed nodes of the created networks created.

The main result is that if 50 nodes with a transmission range of 200 units are placed in a field of 1000×1000 units, there is always at least one fixed node. If there are placed 100 nodes with a transmission range of 200 units or 50 nodes with a transmission range of 250 units, there is a very high probability that there is at least one fixed node. Therefore, in such situations it is not useful to do research for optimizing the MPR strategy, as the network lifetime is in principle determined by these fixed nodes according to Corollary 7.3.2.

As expected, the number of fixed nodes depends on the the number of nodes in the squared field. The more nodes in the field, the more connected a node is and therefore the more choices a node has to reach a 2-hop neighbor. However, if a node has many connections, so a high degree, it does not directly mean that a node can not be a fixed node anymore. For an example, consider the network in which there are two cliques that are connected to each other via one node, which is therefore fixed (cf. Figure 9.1). If we now increase the minimum degree of the nodes in the network, and, thus, also the degree of the fixed node, by adding nodes to the cliques, the fixed node stays fixed.

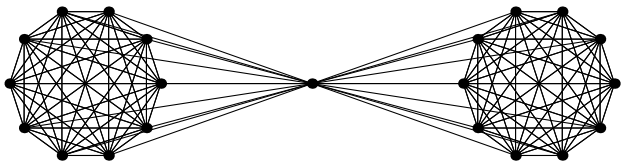


Figure 9.1: An example of a topology in which increasing the degree of the nodes does not effect the numbers of fixed nodes in a network.

9.2.2 Scalability of Fixed Nodes

In the previous subsection we only used one field size. However, it is interesting to know if the number of fixed nodes scales linearly when the area of the field and the number of nodes are scaled linearly. If so, which one intuitively assumes, we could use our results for fields of general size.

		Number of Nodes							
		50	**100**	**150**	**200**	**250**	**300**	**350**	**400**
Range	**200**	11.75	4.13	1.11	0.25	0.06	0	0	0
	250	4.17	0.47	0.04	0	0	0	0	0
	300	1.20	0.01	0	0	0	0	0	0
	350	0.27	0	0	0	0	0	0	0

Table 9.1: The mean number of fixed nodes of 100 generated networks with corresponding network creation parameters.

		Number of Nodes							
		50	**100**	**150**	**200**	**250**	**300**	**350**	**400**
Range	**200**	3.45	2.18	1.14	0.59	0.24	0	0	0
	250	2.24	0.78	0.20	0	0	0	0	0
	300	1.11	0.10	0	0	0	0	0	0
	350	0.65	0	0	0	0	0	0	0

Table 9.2: The standard deviation of the number of fixed nodes of 100 generated networks with corresponding network creation parameters.

		Number of Nodes							
		50	**100**	**150**	**200**	**250**	**300**	**350**	**400**
Range	**200**	5	0	0	0	0	0	0	0
	250	0	0	0	0	0	0	0	0
	300	0	0	0	0	0	0	0	0
	350	0	0	0	0	0	0	0	0

Table 9.3: The minimum number of fixed nodes of 100 generated networks with corresponding network creation parameters.

		Number of Nodes							
		50	**100**	**150**	**200**	**250**	**300**	**350**	**400**
Range	**200**	20	9	6	3	1	0	0	0
	250	9	4	1	0	0	0	0	0
	300	5	1	0	0	0	0	0	0
	350	3	0	0	0	0	0	0	0

Table 9.4: The maximum number of fixed nodes of 100 generated networks with corresponding network creation parameters.

Generation

To study the scalability we use 5 networks settings a create 100 networks for each setting. We keep the transmission range fixed at 200 units and scale the network and the number of nodes in each network. For the first type of networks we position 50 nodes in a field of 1000×1000 units, for the second 100 nodes in a field of $\sqrt{2}(1000) \times \sqrt{2}(1000)$, for the third 150 nodes in a field of $\sqrt{3}(1000) \times \sqrt{3}(1000)$, et cetera. By this we scale the area the number of nodes equally.

Results

The results are presented in Table 9.5 in which A denotes a squared field of 1000×1000 units and nA a squared field with an area of $n \cdot 1000^2$ units. The table shows that if we multiply the number of nodes and the length of the sides of a squared field with a factor α, then in general the mean, minimum and maximum number of fixed nodes will also be scaled with a factor α and the standard deviation will be scaled with $\sqrt{\alpha}$. So, we can conclude that the fraction of fixed nodes in a network depends linearly on the ratio of the area to the number of nodes, as expected.

Note that we scaled the area in the simulations while keeping the form exactly the same. If we do change the form of the area, the number of nodes would not scale that nice anymore. Assume for example that we change the form to a rectangle with one size equal to 1 unit. Then, the nodes will be placed in a line, leading to an increased number of fixed nodes.

	Number of Nodes; Field Size				
	50; A	**100;** $2A$	**150;** $3A$	**200;** $4A$	**250;** $5A$
Mean	11.28	22.37	33.16	45.58	56.81
SD	3.06	4.15	5.08	7.24	6.68
Minimum	6	11	19	28	41
Maximum	19	34	51	65	73

Table 9.5: The effect of increasing the field size on the number of fixed nodes. The term 'SD' denotes the standard deviation.

9.2.3 Maximum Forcedness Ratio of a Network

The Maximum Forcedness Ratio of a network describes 'how fixed' the 'most fixed' node is. This is a property of a network and therefore we have used exactly the same setting as in Section 9.2.1. As a consequence, the results on the Maximum Forcedness Ratio can be related fairly to the results on the number of fixed nodes.

Results

Tables 9.6- 9.9 show respectively the mean, the standard deviation, the minimum and the maximum of the Maximum Forcedness Ratios of 100 generated networks. Similar to the number of fixed nodes, the Maximum Forcedness Ratio decreases if more nodes are placed in the field or if the nodes are assigned

a bigger transmission range. This fits the intuitive idea that if a node has more neighbors, it has more possibilities to reach its 2-hop neighborhood and therefore it is likely that less nodes are forced to be MPR.

The connection between the number of fixed nodes and the Maximum Forcedness Ratios, as mentioned earlier, can be seen easily by looking at the relation between Table 9.3 and Table 9.8 and the relation between Table 9.4 and Table 9.9.

9.2.4 MPR Selection in Graphs on Grids

Theorem 8.4.1 in Chapter 8 basically states that every 2-hop central node in a network chooses for every MPR selection algorithm exactly the same MPR-set. So, the MPR selection algorithms perform identical on the 2-hop central nodes. Therefore one would expect that the network lifetimes of MPR flooding in graphs $G_{m\times n}(r)$ in which the nodes have equal initial battery capacity and transmission costs are also identical or almost identical if different MPR selection algorithms are used. However, there is no theorem for the border nodes such that the effect of the border nodes can be estimated. Because of this, we do simulations on graphs on grids in this section to get a feeling for the border effects.

Simulation 1

We create squared graphs $G_{m\times m}(r)$, in which m is a number between 8 and 15 and r between 1 and 3. Since we do not create random networks, it is sufficient to simulate for each setting only once the effect of the MPR selection algorithm on the network lifetime.

Results

In Table 9.10 the Maximum Forcedness Ratios of the generated grid networks are presented. For a transmission range $r = 1$ of the nodes, the Maximum Forcedness Ratio equals one for each network, which means that in each network there is at least one fixed node. As a consequence, these fixed nodes determine the network lifetime of the networks (see Corollary 7.3.2). Due to Theorem 8.4.1, one might expect that for the generated networks with transmission ranges $r = 2$ and $r = 3$ the network lifetime of MPR flooding is also the same of almost the same if different MPR selection algorithms are used. However, there is dispersion in the network lifetime, as one can see in Table 9.11 where the standard deviation of the network lifetimes reached by MPR flooding with MinCar, MaxWill and MaxWillMinForced is given. The corresponding mean network lifetimes are given in Table 9.12. Since we know by Theorem 8.4.1 that the 2-hop central nodes have exactly the same MPR-set for the different MPR selection algorithms, the dispersion in network lifetime is completely due to the border nodes. We investigate this more in the next simulation.

Simulation 2

To support the rather remarkable conclusion from the previous simulation that the border nodes have much effect on the difference in network lifetimes for different MPR selection algorithms, we carry out some new simulations. Instead

		Number of Nodes							
		50	**100**	**150**	**200**	**250**	**300**	**350**	**400**
Range	**200**	1	1.00	0.97	0.85	0.75	0.66	0.60	0.56
	250	1.00	0.89	0.73	0.60	0.52	0.45	0.41	0.36
	300	0.96	0.66	0.53	0.42	0.36	0.32	0.30	0.27
	350	0.78	0.50	0.37	0.31	0.27	0.24	0.22	0.20

Table 9.6: The mean of the maximum forced ratios of 100 generated networks with corresponding network creation parameters.

		Number of Nodes							
		50	**100**	**150**	**200**	**250**	**300**	**350**	**400**
Range	**200**	0	0.02	0.05	0.10	0.12	0.08	0.11	0.08
	250	0.03	0.11	0.12	0.10	0.08	0.08	0.08	0.05
	300	0.07	0.12	0.13	0.10	0.06	0.06	0.05	0.04
	350	0.16	0.12	0.08	0.06	0.05	0.04	0.04	0.04

Table 9.7: The standard deviation of the maximum forced ratios of fixed nodes of 100 generated networks with corresponding network creation parameters.

		Number of Nodes							
		50	**100**	**150**	**200**	**250**	**300**	**350**	**400**
Range	**200**	1	0.89	0.80	0.62	0.50	0.47	0.42	0.4
	250	0.83	0.6	0.49	0.42	0.35	0.33	0.29	0.26
	300	0.64	0.39	0.32	0.27	0.24	0.24	0.21	0.19
	350	0.42	0.28	0.25	0.20	0.18	0.16	0.13	0.13

Table 9.8: The minimum of the maximum forced ratios of fixed nodes of 100 generated networks with corresponding network creation parameters.

		Number of Nodes							
		50	**100**	**150**	**200**	**250**	**300**	**350**	**400**
Range	**200**	1	1	1	1	1	0.88	0.97	0.75
	250	1	1	1	0.95	0.70	0.85	0.76	0.54
	300	1	1	0.93	0.88	0.64	0.52	0.45	0.42
	350	1	0.96	0.59	0.55	0.41	0.41	0.35	0.29

Table 9.9: The maximum of the maximum forced ratios of 100 generated networks with corresponding network creation parameters.

		Grid Size							
		8×8	9×9	10×10	11×11	12×12	13×13	14×14	15×15
Range	**1**	1	1	1	1	1	1	1	1
	2	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
	3	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29

Table 9.10: The Maximum Forcedness Ratios of the grids.

		Grid Size							
		8×8	9×9	10×10	11×11	12×12	13×13	14×14	15×15
Range	**1**	0	0	0	0	0	0	0	0
	2	18.93	15.71	15.37	14.29	14.73	14.43	10.69	16.77
	3	31.82	29.16	20.88	20.23	17.04	17.62	17.35	16.74

Table 9.11: The standard deviation of the network lifetimes reached by using MPR flooding with the MinCar, MaxWill and MaxWillMinForced MPR selection algorithms for the grids.

		Grid Size							
		8×8	9×9	10×10	11×11	12×12	13×13	14×14	15×15
Range	**1**	20	20	20	20	20	20	20	20
	2	51.67	49.00	46.67	48.67	48.00	47.67	45.33	47.33
	3	56.67	53.67	44.00	43.33	39.67	40.33	40.00	39.33

Table 9.12: The mean of the network lifetimes reached by using MPR flooding with the MinCar, MaxWill and MaxWillMinForced MPR selection algorithms for the grids.

of taking grids with borders, we simulate MPR flooding on the torus variant of grids. In Figure 9.2 a torus is displayed. The main feature of a torus is that every node on the torus is a 2-hop central node. So, there are no border nodes. The torus variant of a grid is created by defining the distance between the first and last column and first and last row both to be 1, such that the distance between every subsequent row and every subsequent column is 1. On these torus variants of grids we simulate again MPR flooding with different MPR selection algorithms and look at the standard deviation of network lifetimes.

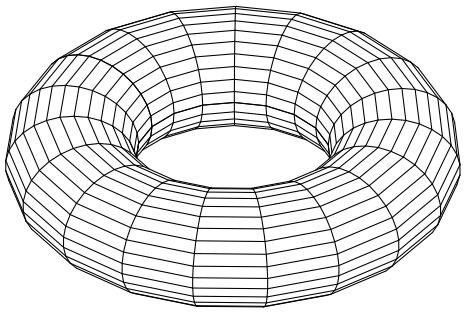


Figure 9.2: An example of a torus.

Results

Table 9.13 shows the dispersion in network lifetimes for different MPR selection algorithms in the torus variants of grids. In Table 9.14 the mean network lifetime of each grid is presented. Again, there are fixed nodes if the transmission range is set to 1 (cf. Table 9.15). But in contrary to Table 9.11, where there is

dispersion for every grid for transmission ranges bigger than one, there are only five tori that have a standard deviation bigger than zero if the transmission range is bigger than one. Furthermore, the values of the standard deviation are much lower for three of the five. These differences in standard deviations between normal grids and the torus variants of grids point out that the border nodes have a large effect on the difference in network lifetimes.

The reason that there are still some performance differences in a torus, is explained for a torus variant of the graph $G_{6\times6}(2)$. This graph is not generated, but is suitable for explaining the concept. Assume that in this graph node $(3,3)$ initiates a broadcast. We now look only at column 3. By one transmission all nodes in that column receive the message, except for node $(6,3)$. To reach also this node an MPR is needed and the MPR can be selected freely from the four neighbors of $(3,3)$. Each algorithm chooses this MPR according to its own discrimination rule, which can lead to different MPR-sets for $(3,3)$. This is the principal idea behind the possible network lifetime differences for multiple MPR selection algorithms in torus variants of grids.

		Grid Size							
		8×8	9×9	10×10	11×11	12×12	13×13	14×14	15×15
Range	**1**	0	0	0	0	0	0	0	0
	2	17.90	0	0	0	0	0	0	0
	3	20.21	0	5.77	2.31	5.20	0	0	0

Table 9.13: The standard deviation of the network lifetimes reached by using MPR flooding with the MinCar, MaxWill and MaxWillMinForced MPR selection algorithms for the torus variants of the grids.

		Grid Size							
		8×8	9×9	10×10	11×11	12×12	13×13	14×14	15×15
Range	**1**	20	20	20	20	20	20	20	20
	2	59.33	63.00	80.00	36.00	76.00	50.00	68.00	39.00
	3	136.67	112.00	83.67	46.33	39.00	43.00	49.00	43.00

Table 9.14: The mean of the network lifetimes reached by using MPR flooding with the MinCar, MaxWill and MaxWillMinForced MPR selection algorithms for the torus variants of the grids.

		Grid Size							
		8×8	9×9	10×10	11×11	12×12	13×13	14×14	15×15
Range	**1**	1	1	1	1	1	1	1	1
	2	0	0.33	0.33	0.33	0.33	0.33	0.33	0.33
	3	0	0.14	0.14	0.14	0.14	0.29	0.29	0.29

Table 9.15: The Maximum Forcedness Ratios of the torus variants of the grids.

9.2.5 Forcedness and MPR Performance

In this chapter we introduced the Maximum Forcedness Ratio, which may be useful for analyzing the performances of MPR selection algorithm. The basic idea is that if a node in the network is forced by many of its neighbors, there are obviously often no other nodes that could be an MPR. Consequently, the MPR selection algorithms do not have much influence on the MPR-set and the network lifetime will differ little for different MPR selection algorithms. In this section we test if this theory holds by simulating MPR flooding using three MPR selection algorithms: the Minimum Cardinality MPR selection algorithm, the Maximum Willingness MPR selection algorithm and the algorithm developed in this thesis, the Maximum Willingness and Minimum Forced MPR selection algorithm. See Sections 8.2 and 8.3 for more information on the MPR selection algorithms.

Simulation

Since the network lifetime of networks with equally distributed energy is completely determined by the existence of fixed nodes in the networks (cf. Corollary 7.3.2), we need to create networks that do not have fixed nodes to have performance differences. Furthermore, to analyze the relation between the Maximum Forced Ratio and the performance differences between different MPR selection algorithms, we need such network setting that results in a big range of Maximum Forcedness Ratios. Based on the results in Table 9.3 and Table 9.6 we choose to place 150 nodes in a square field of 1000×1000 units, while selecting for each simulation a transmission range from the set $[200, 250, 300, 350]$ and assigning this range to all nodes. For every combination of number of nodes and assigned transmission range we create 200 networks, resulting in 800 networks in total.

In each of these networks we initiate broadcast messages according to a sendpattern in which all nodes send successively. The messages are broadcasted by MPR flooding, in which the MPRs are selected by different algorithms in each simulation. Therefore, we can compare the performances of the algorithms fairly. In both the MaxWill algorithm as the MaxWillMinForced algorithm we assume the willingness to be equal to the residual energy to avoid 'rounding effects' due to chosen willingness levels.

Results

The difference in performances of the MPR selection algorithms can be expressed by a Performance Ratio defined as the ratio of network lifetimes obtained by the simulation of MPR flooding with the MPR selection algorithms. Since we focus on three MPR selection algorithms, we also have three comparisons between MPR selection algorithms. We discuss them separately.

MaxWill versus MinCar The results of the simulations are presented graphically in Figure 9.3. The Performance Ratio in this graph is defined as the network lifetime using MaxWill divided by the network lifetime using MinCar.

Analyzing the graph we see that if the Maximum Forcedness Ratio approaches 1 the difference in network lifetimes become smaller. This can also

be concluded from Table 9.16 in which the mean and the standard deviation of the Performance Ratio is listed per intervals of the Maximum Forcedness Ratio. The results support the expected effect that the 'more fixed' the 'most fixed' node is, the smaller the performance differences. If the Maximum Forcedness Ratio is smaller, also bigger performance differences occur. However, a small Maximum Forcedness Ratio is not related directly to a bigger Performance Ratio.

In the graph we can also see that in almost every simulation the network lifetime of MPR flooding using MaxWill is larger than the network lifetime of MPR flooding using MinCar.

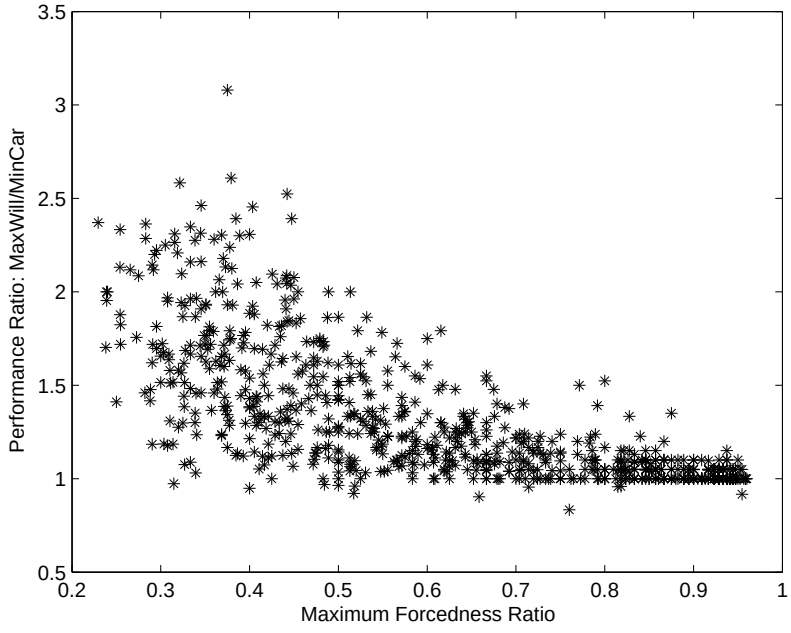


Figure 9.3: The effect of the Maximum Forcedness Ratio on the performance differences using MPR flooding with MPRs selected by the MaxWill algorithm and the MinCar algorithm.

	Maximum Forcedness Ratio Intervals							
	$[0.2, 0.3)$	$[0.3, 0.4)$	$[0.4, 0.5)$	$[0.5, 0.6)$	$[0.6, 0.7)$	$[0.7, 0.8)$	$[0.8, 0.9)$	$[0.9, 1)$
Mean	1.89	1.71	1.52	1.29	1.19	1.10	1.05	1.02
SD	0.32	0.39	0.33	0.22	0.15	0.11	0.06	0.03

Table 9.16: The mean and standard deviation of the Performance Ratio concerning MaxWill and MinCar per interval of the Maximum Forcedness Ratios.

MaxWillMinForced versus MinCar The relation between the performances of MPR flooding using MaxWillMinForced and MinCar to select MPRs are shown in Figure 9.5 and Table 9.17. In the graph the Performance Ratio is defined as the network lifetime reached by using the MaxWillMinForced MPR selection algorithm divided by the network lifetime reached by using the MinCar algorithm. There are some similarities between this graph and the corresponding table and the ones discussed in the previous section, which is not surprising, as the MaxWillMinForced algorithm is based on MaxWill and only adds a sort of look-ahead for the energy consumption. So, the performances vary less if the Maximum Forcedness Ratio approaches one and the MaxWillMinForced algorithm is almost in every situation better than the MinCar.

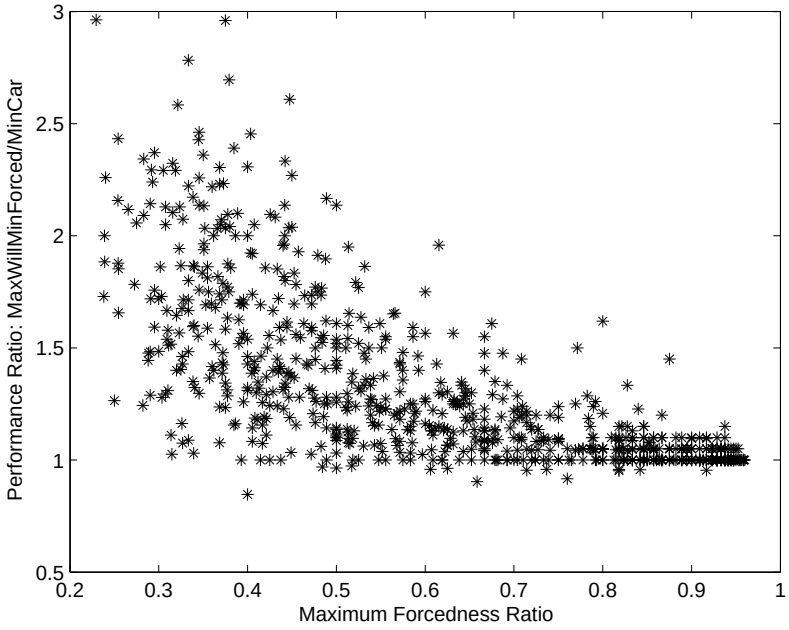


Figure 9.4: The effect of the Maximum Forcedness Ratio on the performance differences using MPR flooding with MPRs selected by the MaxWillMinForced algorithm and the MinCar algorithm.

	Maximum Forcedness Ratio Intervals							
	$[0.2, 0.3)$	$[0.3, 0.4)$	$[0.4, 0.5)$	$[0.5, 0.6)$	$[0.6, 0.7)$	$[0.7, 0.8)$	$[0.8, 0.9)$	$[0.9, 1)$
Mean	1.91	1.74	1.51	1.29	1.18	1.10	1.05	1.01
SD	0.41	0.40	0.34	0.22	0.17	0.11	0.07	0.03

Table 9.17: The mean and standard deviation of the Performance Ratio concerning MaxWillMinForced and MinCar per interval of the Maximum Forcedness Ratios.

MaxWillMinForced versus MaxWill The performance comparison between the MaxWillMinForced and MaxWill MPR selection algorithms is presented in Figure 9.5 in which the Performance Ratio is defined as the network lifetime using MaxWillMinForced divided by the network lifetime using MaxWill. In the graph the effect of the Maximum Forcedness Ratio on the performance differences is less clear compared to the previous graphs. However, if we look at Table 9.18 we see that the standard deviation of the performances decreases if the upperbound of the intervals of width 0.1 is closer to 1. Therefore, also these results support the idea that the performance differences between two MPR selection algorithms is related to the Maximum Forcedness Ratio.

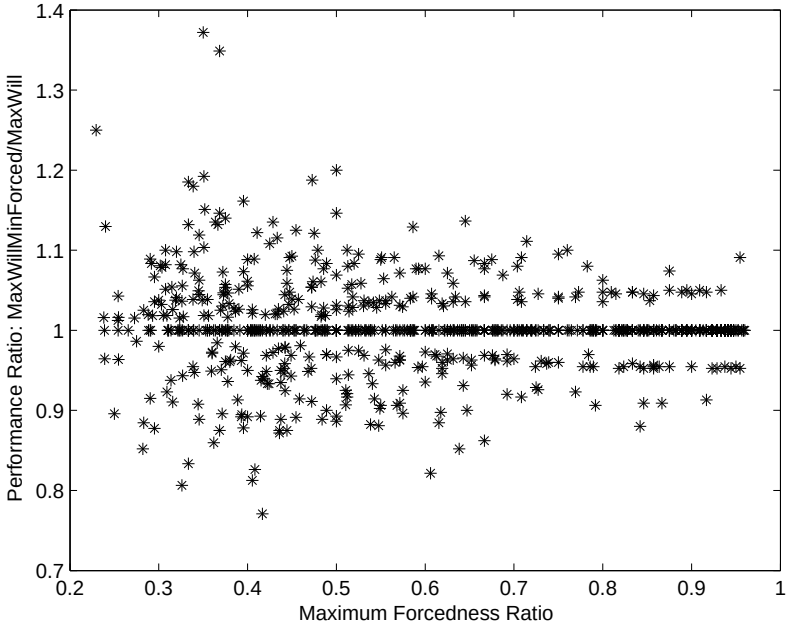


Figure 9.5: The effect of the Maximum Forcedness Ratio on the performance differences using MPR flooding with MPRs selected by the MaxWillMinForced algorithm and the MaxWill algorithm.

	Maximum Forcedness Ratio Intervals							
	$[0.2, 0.3)$	$[0.3, 0.4)$	$[0.4, 0.5)$	$[0.5, 0.6)$	$[0.6, 0.7)$	$[0.7, 0.8)$	$[0.8, 0.9)$	$[0.9, 1)$
Mean	1.01	1.02	1.00	1.00	1.00	1.00	1.00	1.00
SD	0.08	0.08	0.07	0.05	0.05	0.04	0.03	0.02

Table 9.18: The mean and standard deviation of the Performance Ratio concerning MaxWillMinForced and MaxWill per interval of the Maximum Forcedness Ratios.

The algorithm MaxWillMinForced uses the Forcedness Ratio of nodes. This might suggest that there is also a relation between the Maximum Forcedness Ratio and the MaxWillMinForced algorithm. This relation may trouble the

results and could be the reason for a more or less different result compared to the previous results.

When we compare MaxWillMinForced with MaxWill we see that for 424 of the 800 simulations the selection algorithms lead to exactly the same network lifetime. MaxWillMinForced beats MaxWill in 203 simulations, but on the other hand, MaxWill beats MaxWillMinForced in 173 simulations. In Table 9.19 we present a closer look at the simulations in which both algorithms produced the same network lifetime. The mean number of nodes that have a residual energy of less than 5, 10 and 25 are listed for these simulations. Note that we use a packet cost of 5. So, a node with less than 5 units of energy runs out battery if it has to transmit a packet. Therefore, this table gives an indication of how high the probability is that if the node that has died is given a new battery, other nodes run out of energy in the next rounds. But also these results do not give enough arguments to point out a clear winner. The absence of a winner is underlined by the mean of the Performance Ratio, which equals 1.0023.

	MPR Selection Algorithm	
	$MaxWillMinForced$	$MaxWill$
Residual Energy $<$ 5	1.6557	1.6297
Residual Energy $<$ 10	2.4505	2.4646
Residual Energy $<$ 15	3.4976	3.4575

Table 9.19: Comparison for MaxWillMinForced and MaxWill of the mean number of nodes that have a residual energy smaller than 5, 10 or 15 when the first node has run out of energy.

Chapter 10

Conclusion and Recommendations

The introduction of this thesis listed two objectives of the thesis: explore the context of the network lifetime problem and analyze a specific algorithm in this field. These objectives have been reached by classifying the problems and algorithms related to network lifetime in the first part of the thesis and analyzing MPR flooding in the second part. The conclusions that can be drawn from both parts are presented in the first section of this chapter. The next section discusses topics for further research.

10.1 Conclusion

10.1.1 Part I

In the first part of the thesis we discussed the context of network lifetime by classifying and discussing algorithms and problems related to the network lifetime. The classification made clear that each problem needs its own specific approach and that many algorithms are also focused on one specific problem, which is the reason why many algorithms only perform well in the situations for which they have been developed. Besides this, we showed that the performances of the algorithms are strongly related to the sendpattern.

Because of the connection between the performance of an algorithm and the situation in which an algorithm is used, we point out the importance of specifying the details of the problem, network and sendpattern, when doing research on routing algorithms in wireless networks. Therefore also an intensive study is required if one wants to compare algorithms fairly. We did not find such a study in literature, which makes it impossible to select the best algorithm out of the discussed algorithms.

10.1.2 Part II

The center of the research lies in Part II of the thesis. We presented a mathematical analysis of MPR flooding by looking separately at MPRs, MPR flooding

and MPR selection. By this, we could point out the effects and working of the specific elements of MPR flooding. We discuss the most important conclusions.

- We introduced the term fixed node and proved that for networks with equally distributed energy the network lifetime using MPR flooding is completely determined by the fixed nodes, independent of the MPR selection algorithm. By generating networks we obtained results that showed that there is almost for sure a fixed node in a random graph in which there are placed 50 or 100 nodes with transmission range of 200 units in a field of 1000×1000 units. MPR flooding is therefore not favorite for such situations as it has not possibilities to improve the network lifetime. However, the question is whether other algorithms can improve the network lifetime.

  We also showed the scalability of fixed nodes if the form of the field in which the nodes are places remains the same. More precisely, we showed that if the ratio of the area to the number of nodes increases linearly, the fraction of fixed nodes in a network increases also linearly. This indicates that the number of fixed nodes depends only on the area and the transmission range, if the areas have the same shape.

- In Theorem 8.4.1 we proved that for every 2-hop central node $u \in G_{m\times n}(r)$ with $r \geq 1$, u selects $F(u)$ as its MPR-set in a three-step MPR selection algorithm. This basically means that all MPR selection algorithms provide the same MPR-set for 2-hop central nodes in such networks, but may provide different MPR-sets for border nodes. To determine the effect of the border nodes, we simulated grid networks and torus variants of grids and looked at the standard deviation of the performance differences. We could conclude that the borders have a tremendous effect on the differences between the network lifetime for several MPR selection algorithms. Therefore, borders effects can not just be ignored.

- Because of the result that the network lifetime of a network with equally distributed energy and at least one fixed node is independent of the MPR selection algorithm if MPR flooding is used, we had an intuitive idea about the performance differences of MPR selection algorithms in general networks. We assumed that the performance differences in general networks should be related to 'how fixed' the 'most fixed' node was. By simulations, in which we used the Maximum Forcedness Ratio to denote 'how fixed' a node is, we showed that the bigger the Maximum Forcedness Ratio is, the less performance difference occurs. Therefore, it is not worthwile to investigate new MPR selection algorithms if the network has a big Maximum Forcedness Ratio. Furthermore, one should always compare algorithms on exact the same networks. Otherwise, it is possible that one algorithm seems to perform better than another algorithm, but this result may be imposed only since the first algorithm is simulated on networks with a lower Maximum Forced Ratio than the second.

- The last conclusion of our study is about the performances of the three discussed MPR selection algorithm. Without any doubt we can state that in a general stationary network MPR flooding with MPR-sets selected by MaxWill or MaxWillMinForced lead to a longer network lifetime compared

to MPR flooding with the MPR-sets selected by MinCar. However, the comparison between MaxWill and MaxWillMinForced is much more difficult as the simulation results do not point out a winner. We can only say that in some networks MaxWillMinForced is better and in some MaxWill, but that they perform equally in about half of the created networks.

The first three conclusions on the relation between the network structure and MPR selections are important for simulating and comparing MPR selection algorithms. If one does not pay attention to the structure of created networks, it is possible that the network structure is tested instead of the MPR selection algorithm. Therefore one should always investigate if there are fixed nodes in the network that is created and what the Maximum Forcedness Ratio is, such that the results can be compared fairly.

10.2 Recommendations and Further Research

Since, to our knowledge, this is the first analysis of MPR flooding by looking at each step separately, we created the fundament on which new research can be done. In this section we recommend several issues that could be studied further.

- The derived relation between the performance differences of MPR flooding with different MPR selection algorithms and the network structure, raises the need for better algorithm comparison studies in which much attention is payed to the network structure. Specifically, research should be done to compare the algorithms discussed in Chapter 4 fairly.

- In many of the theorems we presented in Part II of the thesis, we assumed the nodes in a network to be positioned on a grid. However, since we only use the translation and symmetry property in the proofs, we believe strongly that the theorems also hold for other regular structures that satisfy the translation and symmetry property, like for example a trigonal structure. We think that even the proofs do not have to be adjusted. But since we did not study this in depth yet, we leave the question open. The presented theorems assume a two dimensional network. Therefore, the mathematical analysis could be extended to higher dimensions. We expect that also for this purpose the proofs do not have to be adjusted or have to be adjusted only slightly.

- Since our simulation study on the effect of border nodes in graphs $G_{m\times n}(r)$ showed that the border nodes are mainly responsible for the dispersion in the network lifetimes of different MPR selection algorithms, we recommend that border nodes should be studied in depth. If the MPR selection can be improved for the border nodes, it is plausible that the network lifetime also might be improved.

- The simulations and theory are based on stationary networks, but in many applications the networks are mobile. We therefore recommend to implement mobility in the simulator and to analyze the aspects that are specifically related to mobility. In Appendix A we already point out how the simulator has to be changed to implement mobility. If mobility is implemented, one should also study a combination of the MaxTotal and

MaxMin algorithms, as stated in Section 4.1.3 where we concluded that such an algorithm would presumably perform very well.

- Other network properties could also be varied to get more insight in the effect of MPR flooding and MPR selection algorithms on the network lifetime. For example, by assigning nodes different initial battery capacities and different transmission ranges. As mentioned in the introduction of Part II the problem we discussed can be characterizes as a SPDR problem. One could therefore also look at dynamic power assignment, but we think that this is a research topic for later. First, the influence of not equally assigned powers should be investigated in a static power assignment context.

- We proposed in this thesis a new MPR selection algorithm, denoted by MaxWillMinForced. The simulations results showed that it can compete with the MaxWill algorithm, but does not beat the algorithm. Further research has to be done to improve the algorithm. First, one should analyze for which networks it performs worse than MaxWill. Hopefully, these networks have a common element that is related to the performance of MPR flooding with MaxWillMinForced MPR-set selection.

- Finally, it would be interesting to study adjustments of the MPR definition. The reason for this is that fixed nodes exist due to this MPR definition that is based on a 2-hop neighborhood. If the definition would be based on a 3-hop neighborhood, probably more paths within this neighborhood would be available between nodes. Consequently, it is likely that there will be fewer forced nodes and, thus, fewer fixed nodes. Even though this adjustment makes the algorithm less local, it may result in longer network lifetimes.

List of Abbreviations

DNLP	Dynamic Network Lifetime Problem
DPDR	Dynamic Power assignment and Dynamic Routing problem
DPSR	Dynamic Power assignment and Static Routing problem
MANET	Mobile Ad Hoc Network
MaxWill	Maximum Willingness MPR selection algorithm
MaxWillMinForced	Maximum Willingness and Minimum Forced MPR selection algorithm
MinCar	Minimum Cardinality MPR selection algorithm
MPR	Multipoint Relay
NLP	Network Lifetime Problem
OLSR	Optimized Link State Routing algorithm
SNLP	Static Network Lifetime Problem
SPDR	Static Power assignment and Dynamic Routing problem
SPSR	Static Power assignment and Static Routing problem
UDG	Unit Disk Graph

List of Notations

uv	An edge between nodes u and v.
$u - v - w$	A path from node u, via node v, to node w.
$d_g(u, v)$	Geometrical distance between nodes u and v.
$F^1(u)$	The set of nodes that are forced by node u to be MPR.
$F^{-1}(u)$	The set of nodes $(f_1, f_2, \ldots, f_n)$ that force node u to be MPR.
$F_2^{-1}(u)$	The set of nodes $(2f_1, 2f_2, \ldots, 2f_n)$.
$C(u)$	The convex hull formed by the extreme points $F(u)$.
$C_2(u)$	The convex hull formed by the extreme points $F_2^{-1}(u)$
$N^k(u)$	The strict k-hop neighborhood of node u, i.e. the set of nodes for which the minimum hop distance to u is k.
$\overline{v}^u$	The node obtained by mirroring node v in node u.
$G(V, E)$	The graph G defined by a set of nodes V and set of edges E.
$G_{m\times n}(r)$	The grid graph with m rows, n columns and in which every node has a transmission range equal to r.
$NLT(G)$	The network lifetime of a network G.
$E_u(t)$	The residual energy at node u at time t.
$C_t(u)$	The energy costs for node u for transmitting a message.
$M(u)$	The MPR-set of a node u.
$MPR(u)$	The set of all possible MPR-sets of u.
$MPR(U)$	The set of all possible combinations of MPR-sets of nodes $u_i \in U$, i.e. $MPR(U) = \{M(u_1) \cup M(u_2) \cdots \cup M(u_m) \mid M(u_i) \in MPR(u_i)\}$.

Appendix A

MPRSimulator

The simulator we developed for simulating MPR selection and MPR flooding in wireless network is programmed in Matlab. The advance of Matlab is that it offers many mathematical functions, but also development tools for profiling and debugging a program. Furthermore, graphical user interfaces can be built in an easy way.

In this section we discuss the basic aspects of our simulator, which is called MPRSimulator. By this we give an idea of how the simulator works and what the possibilities and limitations of the simulator are. To understand the working of MPRSimulator, one should look at the complete code and try the simulator itself.

A.1 Basic Aspects

A.1.1 Type of Simulation

There are generally two types of simulation, time-based simulation and event-based simulation. Time-based simulation works with a clock that checks *each timestep* for tasks and processes them if they are tasks. In event-based simulations the time is not updated each timestep, but at each event. Therefore, all events are processed after each other, with different intermediate time periods. So, a time-based simulator processes the list in which, for example, every second of the simulation is described and an event-based simulator processes the list in which only events are described and jumps, for example, from time is 2 second to 5 seconds and then to 15 seconds.

The advantage of a time-based simulator over an event-based simulator is that it is more intuitive, as the simulated time runs exactly as it would do in real time. However, the main advantages of an event-based simulator over a time-based simulator is that the event-based simulator only visits times in which processing is needed and does not have to choose a timestep, as every moment can be defined and only the order of the moments is important. Because of this advantage, we have developed an event-based simulator.

A.1.2 Events

In the list of events there are only two types of events: selecting an MPR-set and transmitting. Both events have a corresponding ID, time, objective node corresponding to the event and, if necessary, additional information. We describe the events briefly.

MPR-set Selection The node that is related to this event removes its existing MPR-set and selects a new MPR-set, which could be the same as before. In the additional information of the event, it is defined which MPR selection algorithm has to be used to select a MPR-set. After selection of a new MPR-set, all nodes in the MPR-set are instructed to add the selecting node to their MPR Selectors-set. We use the term MPR Selector-set to denote the set of node for which a node is selected as MPR.

Transmitting This event describes the transmission of a message in which the corresponding node in the event is the sender of the message. Note that the sender can be the source of a broadcast message or a relay. As the node transmits the message to its neighbors, it reduces its battery level as it has consumed energy to transmit the message. If the sender has sufficient energy such that the packets arrive at its neighbors, new events are created in which the neighbors of the node transmit the message. Otherwise, the simulation stops and the network lifetime is computed.

A.1.3 Structure of the Simulator

Code Structure

MPRSimulator has a graphical user interface that is separated from the actual simulation code such that the simulator can also run without a graphical interface. This is very useful if many simulations have to be done, as the graphics make the simulator run slower. Since we use an event list, the most important element of the simulator is the event list runner that runs through the list and calls other functions to initiate the processes needed. Each process, like selecting MPRs, is a programmed as a separate program that needs several input variables and produces several result variables.

Lists

The properties of several aspects have to be maintained and updated at certain moments. We have chosen to create specific structures that are filled with arrays, instead of arrays filled with structures, due to efficiency. The most important structures that are used in the simulator are used for each of the following, for the nodes, for the events, for the send statistics and for the received messages. We describe the structures as lists.

List of Nodes This structured list consist of arrays in which a property for each of the nodes is described. The arrays maintain the information concerning the position, power, residual energy 1-hop neighborhood, 2-hop neighborhood, MPR-set $M(u)$, MPR Selector-set, $F(u)$ and $F^{-1}(u)$ of a node u.

List of Events As mentioned earlier, in the event list the following properties of an event are listed: ID, time, objective node, event and additional information (if necessary).

List of Send Statistics The send statistics are used for saving all information of the broadcasts that are simulated. For each broadcast message we save the message ID, source and all times that are related to this broadcast, namely the times that MPRs relay the message and times that nodes receive the message. By this, we can reconstruct the broadcast that occurred during the simulation. We also use the information to compute several results.

List of Received Messages An MPR only relays the first time a message is received. To detect copies of a message, we maintain a list of received messages. In the list, the IDs and sources of received messages are written.

A.2 Possibilities and Limitations of MPRSimulator

A.2.1 Possibilities

Although not the complete functionality of the simulator is used for our research, we do like to mention the strong points of MPRSimulator.

Network Creation For many research questions one likes to have random generated networks that are connected, which is possible in MPRSimulator. However, it is also interesting to look at networks that have non-uniformly random deployed nodes. For example, if one want to simulate the algorithm real situations, where nodes have to be deployed around buildings or lakes. In our simulator, one can load grayscaled images that represent a plan of a lake, a building or a village et cetera. The simulator uses the image for creating the topology by relating the image to a random weighted function. The darker the color of an area, the smaller the possibility that a node will be placed in that area. In black areas there will be placed no nodes at all.

Sendpattern An important aspect of simulations is the possibility to reproduce test result to compare algorithms in a fair way. By creating sendpatterns (a list of times at which nodes broadcast a message) this is possible. Of course these sendpatterns can be created randomly and also can be saved.

Loading and Saving Sometimes a randomly created network that broadcast according to a randomly created sendpattern has a strange behavior that has to be studied further. For these situations it is useful if one can save both the topology and sendpattern. The sendpatterns and topologies can also be loaded to use again for simulation.

Extendable Due to the function structure of the simulator and the comments in the programming code, it is relatively easy to extend the simulator. However, basic knowledge of Matlab is necessary and probably some time to understand the code completely.

A.2.2 Limitations

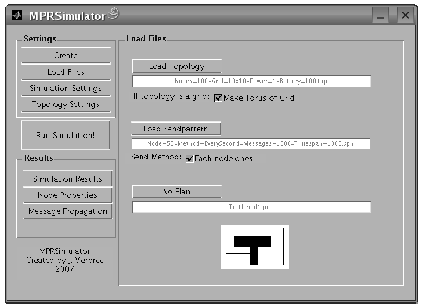
Since the simulator is still in an early stage, it has some limitations. These could of course be solved by programming new functions, but till now this has not been done. We therefore list the actual limitations of the simulator.

Mobility The simulator does not assume nodes to be mobile. Implementation of this aspect is possible, but is not straight forward as we do event-based simulation. So, if a node travels from A to B, we only have the event that the node arrives at B and then its position is updated. During its traveling time, the position property of the node is equal to the position of A. This problem can be solved if for example the random waypoint model is used. When the node has chosen a new destination, for example B, the following properties have to be saved for the node: the departure position (A), the destination position (B) and the movement speed. Also a new event has to be created for the moment that the node arrives at B, in order to create a new movement from B. If now the position of the nodes is asked during his travel from A to B, it can give its departure position, destination position and movement speed such that his position can be computed.

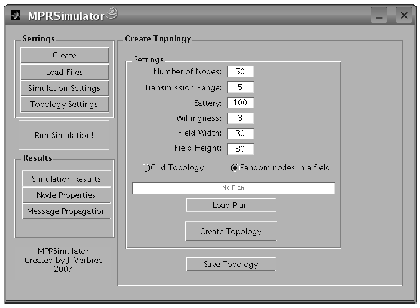
Transmission Models In the simulator we use a rather simple transmission model in. The energy needed for a transmission between node u and v is quadratically related to the distance between u and v. For comparisons of algorithms in in wireless network in the open field this transmission model is not too bad, but for realistic results in cities and forests better models have to be implemented. An useful extension would be a model that takes buildings and forests into account for computing the needed energy for transmission. This could possibly be done in a similar way as the plan by giving a gray scale image in which the darkness defines the power attenuation.

A.3 Screenshots

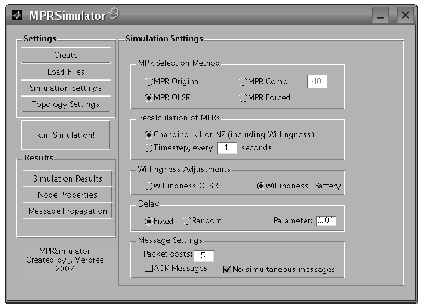
An impression of MPRSimulator is given by the following screenshots. In the caption of the figure comment is given to the several screenshots.



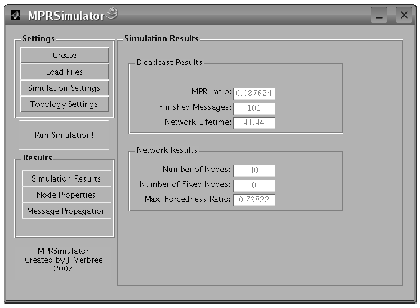
(a) In the file menu one can load topologies, sendpatterns and plans. There are also options to use the torus variant of a grid and to have a sendpattern in which all nodes initiate subsequently a broadcast.



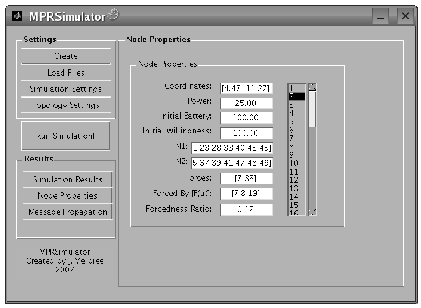
(b) The window for creating a new network. One can set the number of nodes, transmission range, initial battery value, willingness and field size.



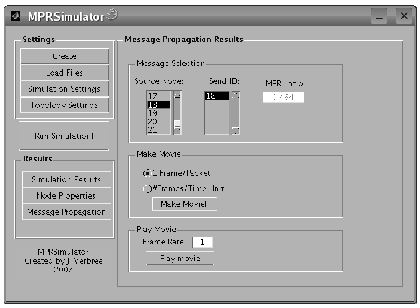
(c) The simulation settings window, in which one can adjust the simulation parameters.



(d) The simulation results windows shows the number of nodes, the network lifetime, the number of messages broadcasted, the ratio of used MPRs over the number of nodes, the number of fixed nodes and the Maximum Forcedness Ratio.

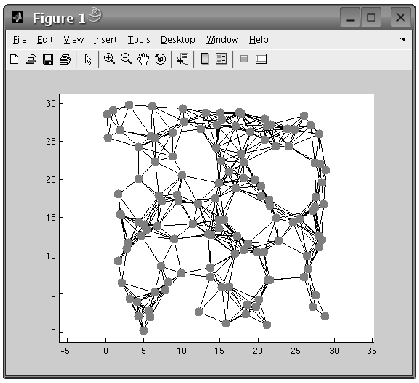


(e) The screen in which the node properties are presented. For each node u the coordinates, power, initial battery, initial willingness, $N(u)$, $N^2(u)$, $F^1(u)$, $F^{-1}(u)$ and Forcedness Ratio is displayed.

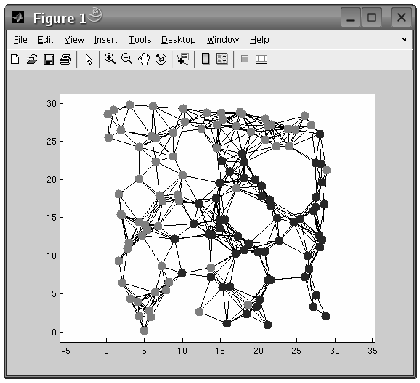


(f) Message propagation can be showed for all messages from the broadcast sendpattern. In this windows one can select the message and adjust some parameters.

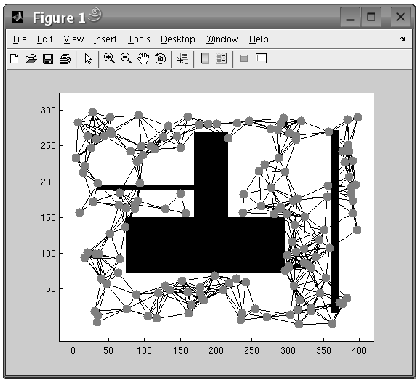
Figure A.1: Screenshots of the windows of the MPRSimulator.



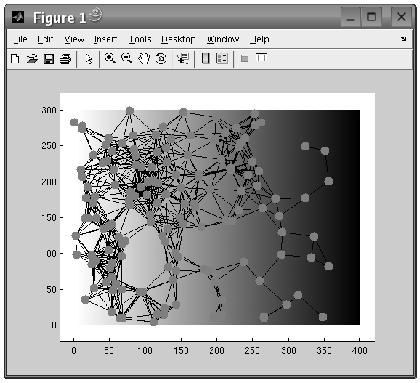
(a) A connected network created by the simulator.



(b) The simulation of a message propagating through a network.



(c) A created network based on a plan, such that there are no nodes in the black areas.



(d) A network created by the simulator based on a grayscale plan. The darker the area, the less nodes that are placed in that area.

Figure A.2: Screenshots of networks created with the MPRSimulator.

Appendix B

MANET Simulators

Selecting a good simulator is often a difficult task. However, there are some papers and websites that compare simulators, but only disappointing little. In [14] four simulators (OPNET, GloMoSim/QualNet, ns-2 and OMNeT++) are described extensively and compared. A short description of some simulators can be found in [7] and a more extensive study on simulators in [12], which discusses nine simulators and four emulators. Lately, a new simulation tool comparison table [16] appeared on the internet that lists the features of the several simulation tools. The now published version is however far from complete.

In this section we discuss only five simulators that may be could be used for MPR-selection and MPR flooding. As MPR-flooding is used in OLSR, we look especially for a simulator that could simulate OLSR. Therefore we describe of each of the five simulator if OLSR is supported. The simulator we discuss are GloMoSim/QualNet, ns-2, OMNeT++, OPNET and Prowler/JProwler. These simulators are selected because of their popularity in the wireless network research area or our knowledge of their programming language. More information on these algorithms can be found in the above listed literature on simulator comparison. We refer the reader to Table B.1 for a brief comparison on several aspects for the discussed simulators.

Simulator	**ns-2**	**GloMoSim (QualNet)**	**OMNeT++**	**OPNET**	**Prowler/ Jprowler**
Programming language	C++, OTcl	Parsec (C, C++)	C++	C, C++	Matlab/ Java
Battery Capacity	yes	yes	yes	yes	no
Power Assignment	yes	yes	yes	yes	no
Transmission models	yes	yes	yes	yes	yes
Mobility models	yes	yes	yes	yes	no
OLSR supported	yes	yes	no	yes	no

Table B.1: Several features of the discussed simulators.

B.1 GloMoSim/QualNet

The free simulation tool GloMoSim ([19], [58]) has been developed in 1998 and has released updates till 2000. After 2000 the software packet has been updated as a commercial product called QualNet ([43]). GloMoSim is based on Parsec, an extension of C for parallel programming. Many standards are supported for multiple layers, including the popular routing maintenance protocols AODV, DSR and OLSR, which is due to the extensibility of GloMoSim. The documentation is a bit poor and the support is stopped, but there is a large user group as it is the second most popular simulation tool used in the sensor network research community, according to [14]. An additional drawback is the uncapability to simulate network types other than IP.

B.2 ns-2

The most popular simulation tool is ns-2, which is the successor of the in 1998 developed Network Simulator (ns). It is based on two programming languages: C++ mainly for implementing protocols and OTcl for the simulation environment itself. There are several reasons for its popularity. First, the software is available for free. Next, ns-2 is designed to be extendible and many researchers used this property to implement their protocol and simulate it with ns-2. The last reason is its popularity itself: if one has an improvement for an algorithm that has been tested earlier with ns-2, it is also the most appropriate simulator to use to show the improvements of the new algorithm. Many protocols are also supported in ns-2, including OLSR. The drawbacks of using ns-2 are the complexity of implementing new protocols and the poor documentation. The fact that two programming languages are used makes the debugging process far more complex, as one needs to have knowledge of both C++ and OTcl. The available documentation is also often outdated, but some support can be found at the large user group.

B.3 OMNeT++

The development of OMNeT++ ([38], [35]) started in 1998 and the simulator is still being updated. The design of this open-source simulation tool resulted in an organized, flexible and easy to use simulator, compared to the other simulators. OMNeT++ is built in C++ and uses modules as building blocks, which simplifies the implementation of other protocols. This component-based method is also the reason for being a faster simulator than ns-2. Other advantages of OMNeT++ are the adjustments that can be done: all layers of the protocol stack can be modified. Because of the organized structure and the possibilities OMNeT++ is qualified as a very good simulator in many simulator comparison papers. However, also a major drawback is mentioned: its popularity. Not many research papers are published using OMNeT++ and important network models are still missing, including OLSR.

B.4 OPNET

The abbreviation OPNET stands for the Optimized Network Engineering Tools algorithm ([39], [10]). This commercial tool is being developed for more than 15 years and is used by large companies and universities. The software is available for free for universities in a country qualifying for free academic licenses. Both the tool and the models are programmed in C and C++. Advantages of OPNET are the included model library and its graphical interface that makes it easy to set up a simulation. A major drawback is that defining new models is not possible without contacting the modeling service of OPNET (according to [14]) and the fact that there are little new protocols already available. However, researches have used OPNET for simulating OLSR. Another possible drawback is the simulation outcome of the flooding tests reported in [9]. Compared to ns-2 and GloMoSim, OPNET gives a significantly different result. Problem is however, that it is not sure whether OPNET or ns-2 and GloMoSim give the right outcome.

B.5 Prowler/JProwler

Prowler ([41], [46]) is a probabilistic wireless network simulator written in Matlab. The equivalent of Prowler written in the programming language Java is called JProwler. The specific goal of the event-driven simulator is to simulate communications in an ad hoc network. A simplified model of the MAC layer is implemented as a plug-in. Also the radio definitions (like propagation) are plug-ins, which makes Prowler suitable for adding new layer and transmission models. The nondeterministic nature of the radio propagation is modelled by a probabilistic radio channel model. Although Prowler is mainly a probabilistic simulator, it can also operate in a deterministic mode to get replicable results while testing a specific application. The Prowler simulator is generic applicable, but targets at Berkely MICA sensor nodes running Tiny OS. Advantages are the extensibility by adding plug-ins and the Matlab environment, which makes it easier to implement new algorithms and analyze the results. The major drawbacks are the lack of plug-ins, including OLSR, support and popularity.

Bibliography

[1] M. Bahramgiri, M. Hajiaghayi, and V. S. Mirrokni. Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks. *Wireless Networks*, 12(2):179–188, 2006.

[2] C. Bettstetter. Smooth is better than sharp: A random mobility model for simulation of wireless networks. *Proceedings of the 4th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 19–27, 2001.

[3] F. Bian, A. Goel, C.S. Raghavendra, and X. Li. Energy-efficient broadcasting in wireless ad hoc networks: lower bounds and algorithms. *Journal of Interconnection Networks*, 3(3/4):149–166, 2002.

[4] D. M. Blough, G. Resta, and P. Santi. A statistical analysis of the long-run node spatial distribution in mobile ad hoc networks. *Proceedings of the International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 30–37, 2002.

[5] M. Cagalj, J. . Hubaux, and C. Enz. Minimum-energy broadcast in all-wireless networks: Np-completeness and distribution issues. *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, pages 172–182, 2002.

[6] G. Calinescu, S. Kapoor, A. Olshevsky, and A. Zelikovsky. Network lifetime and power assignment in ad hoc wireless networks. *Lecture Notes in Computer Science*, (2832):114–126, 2003.

[7] T.W. Carley. Sidh: A wireless sensor network simulator. Technical Report TR 2005-88, Institution for Systems Research (ISR), 2005.

[8] J. Cartigny, D. Simplot, and I. Stojmenovic. Localized minimum-energy broadcasting in ad-hoc networks. *Proceedings - IEEE INFOCOM*, 3:2210–2217, 2003.

[9] D. Cavin, Y. Sasson, and A. Schiper. On the accuracy of manet simulators. *Proceedings of the ACM International Workshop on Principles of Mobile Commerce*, pages 38–43, 2002.

[10] X. Chang. Network simulations with opnet. *Winter Simulation Conference Proceedings*, 1:307–316, 1999.

[11] T. Clausen, P. (editors) Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized link state routing protocol (olsr). RFC 3626, October 2003. Network Working Group.

[12] D. Curren. A survey of simulation in sensor networks. Protect report for subject CS580, http://www.cs.binghamton.edu/kang/david.pdf.

[13] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, and A. Gray. Minimum power broadcast trees for wireless networks: Integer programming formulations. *Proceedings - IEEE INFOCOM*, 2:1001–1010, 2003.

[14] G.A. Di Caro. Analysis of simulation environments for mobile ad hoc networks. Technical Report IDSIA-24-03, Dalle Molle Institute for Artifical Intelligence, Switzerland, December 2003.

[15] O. Egecioglu and T. Gonzalez. Minimum-energy broadcast in simple graphs with limited node power. In *Proc. IASTED Int. Conf. on Parallel and Distributed Computing and Systems, (Anaheim, Canada, 2001)*, pages 334–338, 2001.

[16] CRUISE (CReating Ubiquitous Intelligent Sensing Environments). Simulation tool comparison matrix. http://mobilesummit2006.telecom.ece.ntua.gr:8080/cruise/Public documents/wp123-wsn-simulation-tool-knowledgebase/simulation-tool-comparison-matrix.

[17] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Geometric spanner for routing in mobile networks. pages 45–55, 2001.

[18] N. Ghanem, S. Boumerdassi, and E. Renault. New energy saving mechanisms for mobile ad-hoc networks using olsr. In *PE-WASUN '05: Proceedings of the 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 273–274, New York, NY, USA, 2005. ACM Press.

[19] GloMoSim. Global mobile information systems simulation library, http://pcl.cs.ucla.edu/projects/glomosim.

[20] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[21] I. Kang and R. Poovendran. Broadcast with heterogeneous node capability. *GLOBECOM - IEEE Global Telecommunications Conference*, 6:4114–4119, 2004.

[22] I. Kang and R. Poovendran. Maximizing network lifetime of broadcasting over wireless stationary ad hoc networks. *Mobile Networks and Applications*, 10(6 SPEC. ISS.):879–896, 2005.

[23] K. Kar, M. Kodialam, T. V. Lakshman, and L. Tassiulas. Routing for network capacity maximization in energy-constrained ad-hoc networks, 2003.

[24] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. PLNM, 1972.

[25] L. Li, J. Y. Halpern, P. Bahl, Y. . Wang, and R. Wattenhofer. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. *Proceedings of the Annual ACM Symposium on Principles of Distributed Computing*, pages 264–273, 2001.

[26] X.-. Li. Algorithmic, geometric and graphs issues in wireless networks. *Wireless Communications and Mobile Computing*, 3(2):119–140, 2003.

[27] X.-. Li, G. Calinescu, and P.-. Wan. Distributed construction of a planar spanner and routing for ad hoc wireless networks. volume 3, pages 1268–1277, 2002.

[28] X.-. Li, I. Stojmenovic, and Y. Wang. Partial delaunay triangulation and degree limited localized bluetooth scatternet formation. *IEEE Transactions on Parallel and Distributed Systems*, 15(4):350–361, 2004.

[29] X.-. Li, P.-. Wan, and Y. Wang. Power efficient and sparse spanner for wireless ad hoc networks. *IEEE Int. Conf. on Computer Communications and Networks (ICCCN01)*, pages 564–567, 2001.

[30] X.-. Li, P.-. Wan, Y. Wang, and O. Frieder. Sparse power efficient topology for wireless networks. *Proc. IEEE Hawaii Int'l Conf. System Sciences (HICSS)*, 2002.

[31] W. Liang. Constructing minimum-energy broadcast trees in wireless ad hoc networks. *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 112–122, 2002.

[32] W. Liang and X. Guo. Online multicasting for network capacity maximization in energy-constrained ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(9):1215–1226, 2006.

[33] W. Liang and Y. Liu. On-line disjoint path routing for network capacity maximization in energy-constrained ad hoc networks. *Ad Hoc Networks*, 5(2):272–285, 2007.

[34] C. P. Low and L. W. Goh. On the construction of energy-efficient maximum residual battery capacity broadcast trees in static ad hoc wireless networks. *Computer Communications*, 29(1):93–102, 2005.

[35] C. Mallanda, C. Else, A. Suri, V. Kunchkarra, S. S. Iyengar, R. Kannan, and A. Durresi. Simulating wireless sensor networks with OMNeT++. *IEEE Computer*, 2005.

[36] B. Mans and N. Shreshta. Performance evaluation of approximation algorithms for multipoint relay selection. In *Med-Hoc-Net'04, Bodrum, Turkey*, 2004.

[37] K. Obraczka, K. Viswanath, and G. Tsudik. Flooding for reliable multicast in multi-hop ad hoc networks. *Wireless Networks*, 7(6):627–634, 2001.

[38] OMNeT++ discrete event simulator. hhtp://www.omnetpp.org.

[39] OPNET simulator. OPNET Technologies, Inc., Bethesda, MD, USA. hhtp://www.opnet.com.

[40] J. Park and S. Sahni. Maximum lifetime broadcasting in wireless networks. *3rd ACS/IEEE International Conference on Computer Systems and Applications, 2005*, 2005:1–8, 2005.

[41] Prowler. Probabilistic Wireless Network Simulator. http://www.isis.vanderbilt.edu/Projects/nest/prowler/.

[42] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 9*, page 298, Washington, DC, USA, 2002. IEEE Computer Society.

[43] QualNet Simulator. Scalable Network Technologies, Inc., Culver City, CA, USA. http:/www.scalabale-networks.com.

[44] P. Santi. Topology control in wireless ad hoc and sensor networks. *ACM Computing Surveys*, 37(2):164–194, 2005.

[45] S. Schmid and R. Wattenhofer. Algorithmic Models for Sensor Networks. In *14th International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS), Island of Rhodes, Greece*, April 2006.

[46] G. Simon, P. Volgyesi, M. Maroti, and A. Ledeczi. Simulation-based optimization of communication protocols for large-scale wireless sensor networks. *IEEE Aerospace Conference '03*, 2003.

[47] Y. Tseng, S. Ni, Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, 8(2-3):153–167, 2002.

[48] L. Viennot. Complexity results on election of multipoint relays in wireless networks. Technical Report RR-3584, INRIA, 1998.

[49] P. Wan, G. Calinescu, X. Li, and O. Frieder. Minimum-energy broadcast routing in static ad hoc wireless networks, 2001.

[50] L. Wang and Y. Xiao. A survey of energy-efficient scheduling mechanisms in sensor networks. *Mobile Networks and Applications*, 11(5):723–740, 2006.

[51] R. Wattenhofer. Sensor networks: Distributed algorithms reloaded - or revolutions? *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4056 LNCS:24–28, 2006.

[52] R. Wattenhofer, L. Li, P. Bahl, and Y. . Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. *Proceedings - IEEE INFOCOM*, 3:1388–1397, 2001.

[53] R. Wattenhofer and A. Zollinger. Xtc: A practical topology control algorithm for ad-hoc networks. *Proceedings - International Parallel and Distributed Processing Symposium, IPDPS 2004 (Abstracts and CD-ROM)*, 18:2969–2976, 2004.

[54] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. Resource management in energy-limited, bandwidth-limited, transceiver-limited wireless networks for session-based multicasting. *Computer Networks*, 39(2):113–131, 2002.

[55] J.E. Wieselthier, G. D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. *Proceedings - IEEE INFOCOM*, 2:585–594, 2000.

[56] J.E. Wieselthier, G. D. Nguyen, and A. Ephremides. Algorithms for energy-efficient multicasting in static ad hoc wireless networks. *Mobile Networks and Applications*, 6(3):251–263, 2001.

[57] J. Wu and F. Dai. Mobility-sensitive topology control in mobile ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 17(6):522–535, 2006.

[58] X. Zeng, R. Bagrodia, and M. Gerla. Glomosim: A library for parallel simulation of large-scale wireless networks, 1998.